特集

低消費電力RAM IP化のための設計手法

Design Methodology for Low-Power-Consumption RAM IP

あらまし

ハードIPとしてシステムLSIに搭載されるRAMマクロは，その低消費電力化と高機能化の要求から，論理機能および回路方式はますます複雑になってきている。それにより，RAMマクロをASIC設計手法へ組み込む設計フローはますます複雑化している。それは，ASIC設計手法とRAMのようなハードIPでは，その設計手法が大きく異なるためである。

本稿では，上記問題解決のために開発した，論理モデルと実回路の論理一致検証フロー，またプロセスばらつきと配線クロストークを考慮した遅延ライブラリ作成フロー，さらに電源配線仕様と電源ドロップ解析環境に関して記述する。

本設計手法により，実回路動作とライブラリ間の等価性を高い信頼性で保証できるようになり，RAMをハードIPとして短TATでASIC設計に取り込むことを可能とした。また，RAMマクロのノイズ耐性も同時に保証可能とした。

Abstract

The RAM macros installed in system LSIs now have complicated logical functions and circuits to reduce power consumption and provide high functionality. Furthermore, the LSI design flow for combining these RAM macros with ASICs has become very complicated because the design techniques for hardware IP (e.g., RAM) and ASICs are quite different from each other. To simplify the design flow, we have developed a RAM design methodology that consists of the following steps:

1. A flow for the logical function equivalence check of logical models using actual circuits
2. A delay library generation flow developed by considering process deviation effects and interconnect crosstalk delay
3. A power supply wiring design rule and power-drop verification flow

By using this design methodology, we can minimize differences between the designed library operation and the actual circuit operation. It also enables us to include RAM design (i.e., hardware IP design) into an ASIC design flow with a short turnaround time and solve the RAM macro noise margin problems. This paper describes this methodology.

檜垣直志（ひがき　なおし）

システムLSI開発研究所SOC設計技術研究部 所属
現在，システムLSI物理設計手法に関する研究に従事。

福士　功（ふくし　いさお）

システムLSI開発研究所SOC設計技術研究部 所属
現在，低消費電力化RAMマクロの研究に従事。

笹川隆平（ささがわ　りゅうへい）

システムLSI開発研究所SOC設計技術研究部 所属
現在，低消費電力化RAMマクロとデバイステクノロジ関連の研究に従事。

ま え が き

システムLSI設計における差別化は，有用で高性能な機能を，いかに短TAT（Turn Around Time）で設計するかである。しかし，機能はますます拡大，複雑化しているため，設計および検証作業はますます困難となっている。また，高性能化のためのアルゴリズム，回路的工夫，プロセステクノロジもますます進展しているため，それら新規技術への対応も困難さを増している。

これら広い分野にまたがる技術開発は，LSI設計手法の開発と，高機能IPの開発，の2種類に分類できると考えている。

設計手法とは，例えば以下のような技術である。

(1)　機能記述言語を用いた設計およびエミュレータによるシステムレベルの機能検証

(2)　機能記述言語をもとに，論理合成と自動配置配線ツールによるレイアウト設計手法

もはや機能記述言語（RTL記述）による論理検証，論理合成ツール，自動配置配線ツールを適用しないシステムLSIは考えられない。これら自動化された設計環境を，各設計フェーズ，各設計ブロックに対して有機的に組み合わせ，いかに短TATで複雑高性能なLSIを設計するかが，設計手法の重要な課題である。とくに，高性能と短TATを要求されるLSIでは，この設計手法の善し悪しが大きな差別化要因である。

一方，高機能IPとしては，プロセッサコア，または各種標準バスインタフェースのような論理機能IPと，高速I/Oインタフェース回路，RAMマクロを組み込んだメモリブロックなどのハードIPが挙げられる。これら高機能IPを確保し，短TATでLSIとして組み上げることは，システムLSI差別化のもう1本の柱である。

ところが，自動化された設計手法に高機能IP，とくにハードIPを組み込む場合に，以下のような問題が発生してきている。ASIC設計手法に組み込むための工数がハードIP設計全体の工数に比べて無視できなくなってきている，ASIC設計手法の設計制約のためにハードIPの特性を有効に利用できない，さらに最悪の場合，ハードIPが動作しない，という問題である。これは，ハードIP設計手法がASIC設計手法とは異なり，マニュアル設計であること，スタンダードセルを用いたゲートレベル設計ではなくトランジスタレベル設計であることによりハードIP設計情報をASIC設計手法へ取り込む際に情報の抜けが発生することに起因する。

本稿では，ハードIPを自動化されたASIC設計手法に取り込む場合の課題と，その解決方法に関して，組み込み用プロセッサ[(1)]に搭載されたRAMマクロを例に記述するものである。

自動化設計に取り込む場合の課題

ハードIPをLSIに組み込む場合，以下の緒元を明確にする必要がある。

(1)　インタフェース信号の論理動作

(2)　インタフェース信号の遅延特性

(3)　ノイズ耐性を満たすための使用制限

（1）と（2）に関しては，ASIC設計で用いられるハードIPの機能および特性を正確に反映したライブラリを準備することに集約される。

（1）は，論理ライブラリと実回路との論理等価性の検証，（2）は，遅延ライブラリの作成とその実回路特性との一致確認が必要となる。

ところが，上記ライブラリ作成とその検証作業が困難さを増している。それは，組み込みプロセッサに適応するRAMマクロが，その性能向上を図るために論理機能および回路方式を複雑化させていること，また，ASIC設計手法は様々なツールを組み合わせて実現していることから，多数のライブラリ品種が必要となり，上記ライブラリ作成と検証作業が複雑となってきているためである。そのため，実回路の特性と，ライブラリで表現されている特性にミスマッチが発生する危険性が生じてきている。

（3）の問題は以下のとおりである。RAMマクロライブラリではノイズ耐性を十分表現しきれない。よって，LSI全体のノイズ解析ではRAMマクロ内部のノイズ解析ができない。一方RAMマクロ単独のノイズ解析では，LSIの中でRAMがどのように使われるかを特定できないため，この解析にはLSI全体の実動作を反映できない。そのため，RAMの使用制限を適切に与えることでLSI全体のノイズに対する品質を確保することが必要となる。

以下の章では，上記3種類の問題の詳細と，それらを解決するための設計フローおよびその設計環境に関して，また，発生する工数をいかに削減したかに関して記述する。

論理ライブラリと実回路との論理等価性検証

組み込みプロセッサ用RAMの論理機能が複雑化している例としては，以下が挙げられる。

(1)　低消費電力化のために多数の動作モードを設定

する。

(2) LSI全体の性能向上のためにプロセッサ論理機能の一部を取り込む。

(1)の例として，今回開発した組み込みプロセッサ用のRAMの動作モードを**表-1**に示す。このRAMでは5種類の動作モードを持たせ，細かいパワーセービングを可能とした。

また，(2)の例としては，デュアルアドレスアクセス機能が挙げられる。これは，RAMマクロを2バンクに分け，それぞれ独立にアドレス系統を設定することで効率的なデータアクセスを可能とする機能である。そのほかの代表的なRAM機能としては，キャッシュの有効・無効を指示するバリッドビットのクリア機能が挙げられる。

このように，組み込みプロセッサ用のRAMマクロは，これまでの汎用オンチップRAMマクロに比べ，論理動作が複雑化してきている。このため，RAMマクロ論理検証項目は増加し，その結果，論理ライブラリと実回路との論理等価性検証TAT短縮はますます重要となってきている。

今回著者らは，**図-1**に示す論理等価性検証フローを構築し，上記課題を解決した。Verilog論理シミュレータにCadence社製NC-Verilogを，トランジスタレベル回路シミュレータに，Mentor社製Mach-TAを用いている。RAM論理機能のマスタ情報は，Verilogで記述されたBehaviorモデルである。このモデルはチップ全体のRTLレベル論理検証で用いられるモデルと同一である。RAM機能を検証するためのテストベンチと，このBehaviorモデルとを，NC-Verilogで実行することで，Behaviorモデルの機能検証を行う。RAMマクロ全体のトランジスタレベル論理検証のためには，NC-Verilog実行時の，RAM Behaviorモデルインタフェース部の信号遷移情報を保存し，これを，トランジスタレベル論理検証の入力ベクタおよび出力期待値として用いる。これにより，RAM実回路が論理ライブラリと同一の動作をすることが確認でき，論理ライブラリと実回路の論理等価性が保証される。

今回開発したRAMマクロのうち，最も機能が複雑なData-cache RAMの検証項目数および検証ベクタサイクル数を**表-2**に示す。このRAMマクロは，機能が複雑なことから検証項目が膨大になり，その結果検証ベクタ数も増加した。しかし，今回の設計フローを用いることで，このような膨大な検証ベクタも複数のCPUで実行することで，実質20日で完了することができた。

本設計フローのかぎは，

(1) 340万という膨大なベクタを数週間で実行可能な回路シミュレータ

(2) Verilogシミュレータ環境とトランジスタレベルシ

表-1 Data-cache RAM概要

ポート数	2 RW
メモリ容量	1,024ワード×64ビット
動作モード	スタンバイモード リードモード パーシャルリードモード ライトモード パーシャルライトモード
そのほかの機能	デュアルアドレスアクセス機能

表-2 Data-cache RAM論理検証内容

検証項目数	317
総ベクタ数	3,420,700
総CPU時間	5,590時間/CPU

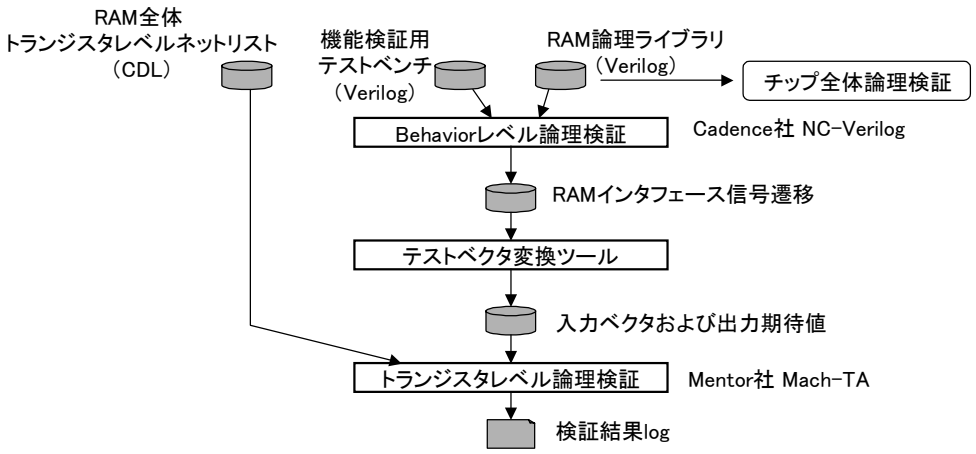


図-1 論理ライブラリと実回路との論理等価性検証フロー
Fig.1-Logical function equivalence check flow.

ミュレータ環境のスムーズなデータ受け渡し方法の構築

にあり，これにより，チップ全体RTLシミュレーションとの一貫した論理検証環境の構築が可能となった。

本フローを用いることで，実際に数項目の障害がRAMマクロ設計初期段階で検出された。よって，LSI全体での論理バグ混入を事前に防ぐことができたと考えている。

遅延ライブラリ作成とその課題

● 遅延ライブラリ作成上の問題点

遅延ライブラリ作成の観点からは，ハードIPをASIC設計フローに組み込む場合，以下を解決する必要がある。

(1) 実回路特性とライブラリ特性の一致保証
(2) 各種ツール用遅延ライブラリの開発TATの削減
(3) プロセスばらつきを考慮した遅延モデル作成

今回の組み込みプロセッサでは，以下の4種類の遅延ライブラリが必要となった。

(1) 論理合成用ライブラリ（dc lib）
(2) 自動配置配線用ライブラリ（TLF lib）
(3) スタティック遅延解析用ライブラリ（CLL2B）
(4) バリデーション用ライブラリ（遅延付きVerilogモデル）

上記4種類のライブラリすべてが，実回路特性を正確に反映し，かつ全ライブラリ間の等価性が補償され，さらにそれらを短TATで開発することが要求される。

また，プロセステクノロジの進展により，以下の課題を解決する必要がある。

(1) 配線カップリング容量によるクロストーク遅延の考慮
(2) プロセスばらつきの反映

以下では，上記問題を解決する設計フローに関して述べる。

● 遅延ライブラリ作成フロー

今回開発したRAMマクロ遅延ライブラリ作成フローを**図**-2に示す。本フローは以下の二つの大きな特徴を持つ。

第1の特徴は，遅延解析用ネットリストとして実デバイスから配線容量/抵抗抽出したRAM全体のトランジスタが含まれるネットリストを用いていることである。これまでの遅延ライブラリ作成フローでは，クリティカルパスのみのSPICEネットリストを用いることが主流であった。しかし，これでは実物の回路特性を反映するのが難しくなってきている。それは，回路方式が複雑になることで各入出力端子間のタイミング依存関係が複雑になり，すべての条件を反映させたクリティカルパスも複雑になること，また配線微細化によりカップリング容量の考慮が必須となってきているが，配線カップリング容量を考慮したクリティカルパスネットリスト作成は困難であることがその理由である。

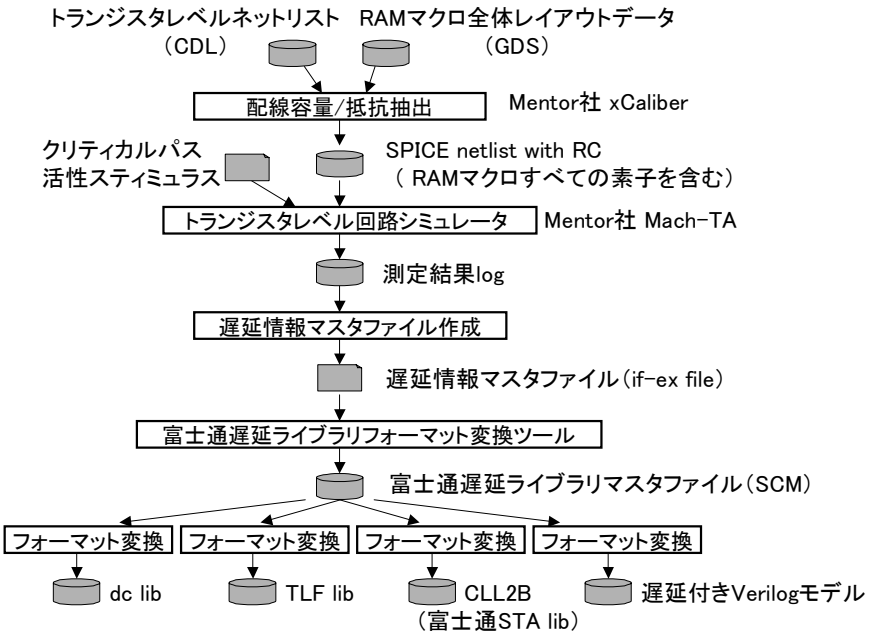


図-2 遅延ライブラリ作成フロー
Fig.2-Delay library generation flow.

本フローでは，回路全体のネットリストをそのまま回路シミュレータに入力しているため，上記のようなクリティカルパス生成時の問題は発生しない。また，すべてのノードでカップリング容量を考慮したネットリストを用いて解析しているため，適切な入力ベクタを作成することによりクロストーク遅延の反映が可能となった。さらに，クロストークノイズによる誤動作検証も可能である。

第2の特徴は，様々の遅延ライブラリを独立に作成するのではなく，すべての遅延情報を含む遅延ライブラリマスタファイルを作成し，それを各種ライブラリごとの変換ツールにより各種ライブラリを作成する点である。共通のファイルから自動生成しているため，各ライブラリ間での遅延特性の一致を保証することが容易となり，かつ複数のライブラリ作成工数の短縮も可能となった。

● プロセスばらつきの考慮

プロセステクノロジの進展により，プロセスばらつきによるトランジスタ特性のばらつきは，ますます大きくなってきている。これまでは，ロット間，またはウエハ間のばらつきを主に考慮してきた。ところが，0.13μmテクノロジ世代では，ウエハ内，さらには，チップ内のトランジスタ特性ばらつきまで考慮する必要がある。RAMマクロ内部回路設計では従来からチップ内ばらつきは考慮されてきたが，今回，遅延ライブラリ作成においても，チップ内ばらつきを考慮する仕組みを取り入れることで，プロセスばらつきへの対応を強化した。

遅延ライブラリを作成するときに，チップ内ばらつきが問題となるのは，セットアップタイムとホールドタイムを定義する場合である。**図**-3に示すように，セットアップタイムとホールドタイムは，クロック系のパス遅延と，データ系のパス遅延の差で定義される。チップ内ばらつきが無視できない場合，それぞれのパス遅延は，回路シミュレータで測定する値に比べて，各トランジスタ特性ばらつきにより独立に変動することとなる。

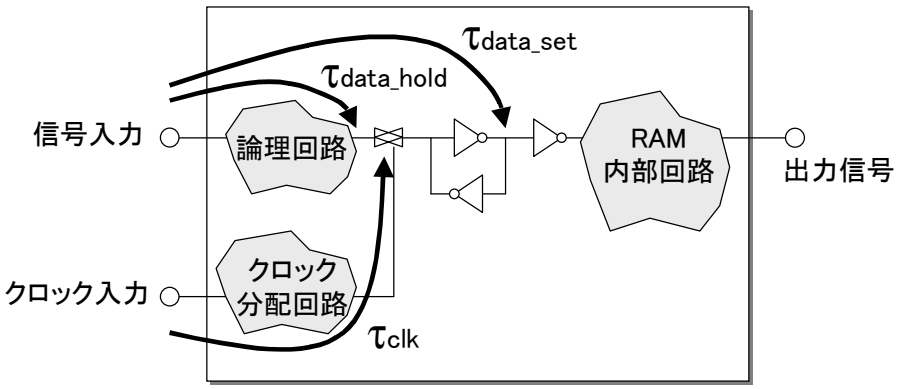
そこで，図-3に示すように，ばらつき効果の1次近似として，クロック系パス遅延と，データパス遅延にそれぞれ独立に係数を掛けてセットアップタイム，ホールドタイムを定義することで，チップ内部プロセスばらつきの効果を反映可能とした。この係数は，前節で述べた遅延情報マスタファイル（if-ex file）内に設けた，ばらつき係数定義フィールドに設定する。富士通遅延ライブラリマスタファイル（SCM）には，上記ばらつき係数を考慮して算出された遅延パラメタが反映される。

この特性ばらつき係数は，以下の考え方により概算できる。使用するプロセステクノロジの特性ばらつき標準偏差σi，遅延パスのトランジスタ段数Nとすると，当該パスの遅延ばらつきの標準偏差σNは以下の式となる。

$$\sigma N = \sigma i / SQRT(N)$$

遅延パスのトランジスタ段数が10段，標準偏差3σでの特性ばらつきが30%とすると，ばらつき係数αは，

$$\alpha = 1 + \sigma N$$



$\tau setup = \tau data\_set - \tau clk / \alpha$

$\tau hold = \tau clk \times \alpha - \tau data\_hold$

$\tau setup$: セットアップタイム

$\tau hold$: ホールドタイム

α : ばらつき係数

図-3　遅延パラメタの定義方法
Fig.3-Delay parameter definition.

＝1＋0.3/SQRT（10）

＝1.0948

となり，チップ内ばらつきにより約10%のマージンを設定することになる。

以上の考え方とライブラリ作成環境を整備することで，プロセスばらつきを考慮したライブラリ作製を可能とした。

ノイズ耐性を満たす設計仕様とノイズ解析

● ノイズ発生要因

現状のASICにおいてノイズの主な原因は，電源ドロップと，配線カップリング容量によるクロストークノイズである。基本ゲートのみを用いたASIC設計においては，その解析手法が構築されつつあるが，ハードIPを含めた上記解析手法は確立されているとはいえない。ハードIPとしてのRAMマクロは，その回路特性をライブラリという形で表現しているため，RAM内部の個々の特性をチップ全体解析に反映できないためである。

そこで，あらかじめ当該ノイズ要因を避けるための適切な設計規約を設定しておく必要がある。

● 電源品質確保のためのレイアウト規約

今回開発した，RAMマクロは**図-4**に示すような電源配線方法を採用した。RAMマクロの信号配線はすべてメタル3層以下で形成し，メタル4層目はすべて電源を短冊状に配置している。

このようなレイアウトトポロジの採用には，以下の利点がある。

小信号振幅回路動作であるためRAMの動作上最もノイズに弱いビット線が，メタル4層でシールドされることとなり，RAMマクロ上を通過する信号配線に起因するカップリングノイズの効果を考慮する必要がなくなる。そのため，RAM上部にLSI全体のブロック間配線を通過させることが可能となる。

またRAMマクロへの電源供給は，マクロ周辺に電源リングを作成させる必要はなく，メタル5層から直接供給可能となる。このため，RAMマクロ内部の電源ドロップを緩和することが可能となった。

最大メタル8層品が可能なプロセステクノロジ開発により，このような電源トポロジを適応することが可能となった。

● 電源ドロップ解析手法

RAMマクロの電源品質を確保するためには，チップ最上位レイアウト設計で実行されるRAMマクロへの電源配線を十分に確保する必要がある。そのためにはRAMに供給する必要最低限の電源供給数をあらかじめ見積もり，電源設計仕様とする必要がある。この目的のために，図-4に示すように，メタル5層以上のLSI電源配線をRAMマクロ上部に配置し，RAMマクロ電源であるメタル4層目に，直接コンタクトする電源トポロジを採用した。またこの電源トポロジを採用した場合のRAMマクロ消費電力を考慮した電源網解析環境を構築し，必要な電源供給量を見積った。

解析は，以下の手順で行った。RAMマクロレイアウトデータから電源抵抗成分を抽出し，電源網のSPICEネットリストを作成する。各ノードには，以下の方法で算出した電流が定義されている電流源を付加する。各セルの消費電流は，その出力ノードに繋がる容量と，その容量が放電される活性化率，およびRAMマクロ動作周波数より求める。またセンスアンプのように出力ノードに繋がる容量に依存しないセルでは，回路シミュレーションにより求めた電流値を直接設定する。このネット

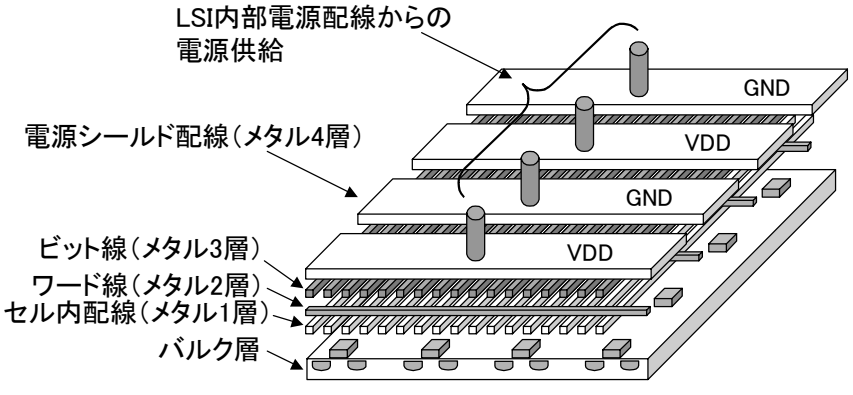


図-4　電源配線方法
Fig.4-Power line implementation.

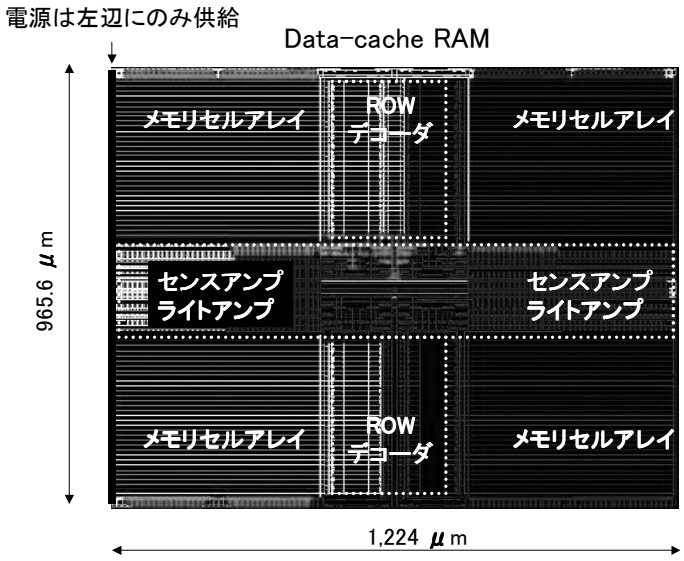


図-5　電源ドロップ解析結果
Fig.5-IR drop simulation result.

リストを用いて回路シミュレーションを実行することで，各電源ノードでの電源ドロップ量を見積った。

図-5は，最上位階層から電源供給される場所をRAMマクロ左辺にのみ与えた場合の電源降下量を示している。白線は，電源ドロップ基準を満たしている電源配線であり，灰色の部分は基準を満たさない電源配線である。このように厳しい条件では，右側のメモリセルアレイ部とセンスアンプおよびライトアンプ配置部の70%程度が電源ドロップ基準を満たしていない。

RAMマクロ上に100μm間隔で電源供給を行った解析では，すべてのRAM内部ノードで電源ドロップ許容値を満たすことが確認された。よって，少なくとも100μm間隔でRAMに電源を供給することをLSI電源仕様とした。以上により，RAM内部の電源ドロップを抑制することが可能となった。

むすび

ハードIPとしてのRAMマクロを，自動化されたASIC設計手法に組み込む際の設計手法に関して述べた。

第一に，チップ全体のRTLレベル論理検証環境と，RAM回路のトランジスタレベル論理検証を，共通のVerilog Behaviorモデルを用いて実行することで，ライブラリと実回路の論理等価性を高いレベルで保証可能とした論理等価性検証環境の構築，2点目は，実レイアウトから抽出したSPICEネットリストを使用すること，およびプロセスばらつきを考慮した解析手法による，実回路特性を正確に反映した遅延ライブラリ作成フローの構築，3点目が，ノイズ耐性を実現するための電源設計仕様の構築とRAM内部電源ドロップ解析環境の構築である。

今回開発した設計環境により，より高性能で複雑な機能を持つRAMマクロを，高い信頼性を持ち，また短TATでシステムLSIに組み込むことが可能となったと考えている。

参考文献

(1) H. Okano et al.：An 8-way VLIW Embedded Multimedia Processor with 7-layers Metal 0.11 μm CMOS Technology . 2002 IEEE International Solid-State Circuits Conference.