

32ビットマイクロプロセッサV810の設計手法

鈴木宏明* 鈴木千佳** 木村晃子** 佐藤庄一郎*** 井手秀一*** 坂中康秀***

*NEC ULSIシステム開発研究所基盤技術開発部
**NEC ULSIシステム開発研究所CAD応用技術部
***NECアイシーマイコンシステム(株)九州LSI開発センター
マイクロコンピュータ第二技術部

*〒211 川崎市中原区下沼部1753
電話 (044) 435-1523
Fax. (044) 435-1753

あらまし

V810 (μ PD70732)はRISC技術を利用した組み込み用途向け32ビットマイクロプロセッサである。0.8 μ m、CMOS2層アルミプロセスを使用し、24万トランジスタを7.7 \times 7.7mm²のシリコンチップ上に集積した。

低電圧動作(2.2V \sim 5.0V)、広い動作周波数(DC \sim 25MHz)、低消費電力動作(25MHz時500mW)を実現した。

自動設計技術の本格的導入により、テープアウト前に、機能、論理、遅延、高故障検出率の達成、等の検証を実行し、ファーストシリコンにおいて、機能、スピード等が設計目標を達成していることを確認した。

和文キーワード CMOS, RISC, マイクロプロセッサ, CAD, 論理合成, 遅延検証

A 32-Bit RISC Microprocessor V810 and it's design techniques.

Hiroaki Suzuki

ADVANCED PROCESSOR ARCHITECTURE
DEVELOPMENT LABORATORY
ULSI SYSTEMS DEVELOPMENT LABORATORIES
NEC Corporation

1753, SHIMONUMABE, NAKAHARA-KU,
KAWASAKI, KANAGAWA, 211 JAPAN

TEL (044) 435-1523 DIRECT
FACSIMILE (044) 435-1753

Abstract

An advanced 32-bit RISC microprocessor for embedded controls; V810 and it's design technique are described in this paper. The V810 is fabricated by using 0.8 μ m CMOS double metal layer process technology to integrate 240,000 transistors on a 7.7 \times 7.7mm² die. In design of the V810, we used design automation techniques. The V810 was analyzed for logical correctness and timing constraint before fabrication. Finally, V810 executed realtime-OS and SPEC benchmarks correctly at first silicons.

英文 key words CMOS, RISC, Microprocessor, CAD, Logic synthesis, Timing verification

1 はじめに

近年マイクロプロセッサの性能の向上は著しく、ワークステーション等の高速化の一翼を担っている。一方パーソナル向けの電子機器は小型、軽量、薄型化が進み、これらの搭載デバイスに対して高性能、低消費電力、低電圧動作、広い動作周波数範囲などの条件が強く求められるようになってきた。

従来マイクロプロセッサの様な汎用デバイスの設計手法はスピード、チップ面積等の理由からフルカスタム設計（マニュアル回路設計、マニュアルレイアウト）が主流であり、最新のデバイステクノロジーを採用し膨大な設計工数を投入、設計期間も長期化する傾向にあった。

近年ASICの需要の高まりにつれてゲートアレイ、スタンダードセル設計が普及し、LSI設計の各フェーズにおいてHDL（Hardware Description Language）による設計、論理合成、自動レイアウト、バックアノテーションといったトップダウン的な自動設計手法が取り入れられている。最近ではツールの進歩により、かなりの部分でフルカスタム設計と遜色無く目的の動作周波数、機能、特性等を保証し設計期間を短縮出来るようになってきた。

今後、回路の複雑化、大規模化、設計の短TAT化を実現するためには、CAD技術の重要性はますます高まっている。

本稿では、V810の概要について触れ、その設計、開発手法とCAD環境を以下に紹介する。

2 V810の概要

V810はRISC技術を採用した組み込み用途向け32ビットマイクロプロセッサである。0.8 μ m CMOS 2層アルミプロセスを使用し、24万トランジスタを7.7 \times 7.7mm²のシリコンチップ上に集積した（写真1）。整数演算性能は15MIPS（25MHz動作時）であり浮動小数点演算性能は約0.89MFL OPS（同）である。またV810は1Kバイトの命令キャッシュを内蔵し、浮動小数点演算機能、ビットストリング機能を有する。

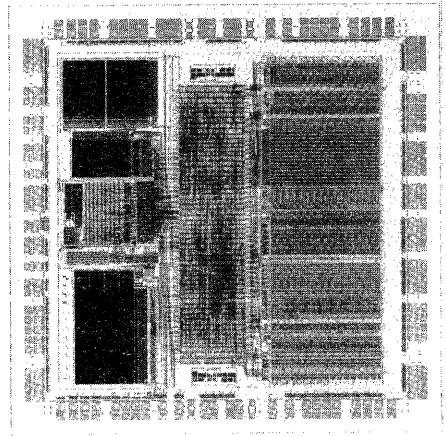


写真 1

3 開発の方針

V810は当初よりプリンタ、FAX、携帯型電子機器等の組み込み用途をターゲットとし、次のような目標を掲げ設計を開始した。

（1）設計期間の短縮化

市場ニーズへの迅速な対応。自動設計手法の導入、確立した半導体プロセスを使用する。

（2）低消費電力、低電圧動作

バッテリー駆動を考慮した動作。低電圧時の回路シミュレーションや回路方式の工夫により実現する。

（3）整数演算性能の向上

RISC技術の採用。5段階からなる命令パイプライン方式をとる。

（4）高いコストパフォーマンス

チップ面積の削減。機能を絞り、トランジスタ数を削減する。

（5）組み込み用途向けアーキテクチャ

CISC的命令を用意。ワイヤードロジック方式とマイクロプログラム制御方式を混在させる。

4 設計方式

V810はデータバス部、マイクロROM RAM部、クロックドライバ部、制御部、命令デコード部、端子部とこれらを接続するブロッ

ク間配線部からなる。以下に各ブロックの概要と特徴を示す。

(1) データバス部

データバス部は、汎用レジスタ、ALU、浮動小数点ハードウェア、バスコントロール部などからなる。

- 繰り返し構造
- ALUのキャリーライン等、明らかなクリティカルバスが存在する
- ダイナミック回路の使用により面積削減可能

(2) マイクロROM, RAM (キャッシュメモリ) 部

- ブロックの大部分がメモリセルの繰り返し構造
- 配線数が多くレイアウト的制約が多い

(3) クロックドライバ

内部で使用する非重複な2相クロックを生成する。

- 温度、製造条件にかかわらず、波形の保証が必要
- 動作周波数に直接影響するためアナログ解析が必要

(4) 制御部

バスシーケンサ、マイクロROM出力デコードを行う部分である。

- 繰り返し構造を持たない
- RTレベルでは(設計者が理解しやすい様)冗長な記述がされている
- クリティカルバスが明らかではない
- 設計の最後まで変更が入り、バグの入りやすい箇所である

(5) 命令デコード部

命令デコードを行いRISC命令については制御信号を生成し、CISC命令についてはマイクロROMのアドレス制御を行う。

- マイクロROM中のファームウェアの変更により回路変更が必要である
- 繰り返し構造とランダムロジックが混在する
- タイミングクリティカルを含む
- レイアウト的制約が大きい

(6) 端子部

- チップ面積に直接影響する
- 繰り返し構造
- 電力の大半を消費しラッチアップ、ESD等に対する考慮等アナログ解析が必要

(7) 各ブロック間配線

- バス、クロックラインを含む
- 配線数が比較的多く規則性がない
- 配線領域がチップ面積のネックとならない

以上のことから、各部の設計方法を(表1)のように決定した。

| ブロック | 回路 | レイアウト |
|--------------|-----------------|----------|
| データバス | マニュアル方式 | マニュアル方式 |
| マイクロROM, RAM | マニュアル方式 | マニュアル方式 |
| クロックドライバ | マニュアル方式 | マニュアル方式 |
| 制御部 | 自動 | 自動 |
| 命令デコード部 | 自動+マニュアル+PLAを使用 | 自動+マニュアル |
| 端子部 | マニュアル方式 | マニュアル方式 |
| ブロック間配線 | マニュアル方式 | 自動 |

表1 各ブロックの設計方式

5 設計フロー

V810の設計、検証フローを(図1)にあげる。設計フロー、検証フロー

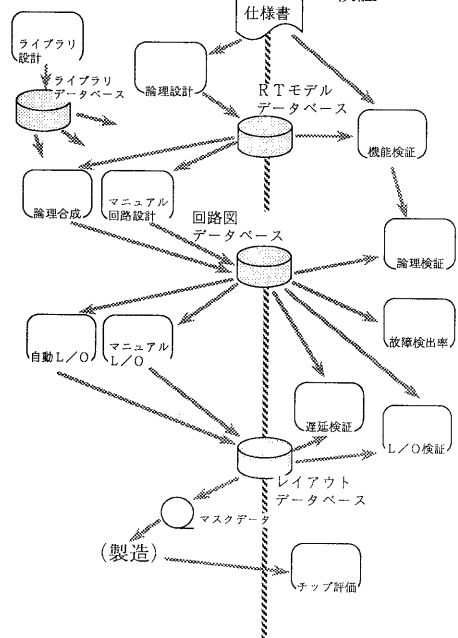


図1 設計、検証フロー

以下に各フェーズの内容を示す。

(1) アーキテクチャ設計

設計開始時にV810の命令セットを実行するアーキテクチャシミュレータをC言語で作成した。このアーキテクチャシミュレータはアプリケーションソフト等を実行し性能評価を行うとともに内製のコンパイラ、アセンブラの評価、RTモデルの検証に用いた。

アーキテクチャシミュレータは仕様変更が容易、シミュレーションスピードが早い、というメリットがあるがハードウェア資源の割り当てや、動作の詳細が分らないなどのデメリットがある。

V810では短期間で設計を達成するため、性能を確認した時点で、RT設計に入り並行して回路設計を開始した。

(2) RT設計

アーキテクチャ検討の結果に基づきレジスタトランスファ(RT)レベルの機能設計を行なった。RT設計は実チップと詳細な点まで同一の動作を行なうシミュレーションモデルを作成し機能の検証を行なった。

RTモデルはハードウェア割り当てを強く意識した記述を行ない、回路設計者に対して回路構成のヒントを与え、またチップ面積やタイミング等の見積を可能にした。

RT検証用テストボタンは基本命令テスト、例外割り込み等のテストをシステム設計者が作成し正当性を検証した。リアルタイムOSを実行し、またテストボタン作成ツールによる命令のランダム組合せテストボタンを実行し結果の照合を自動的に行なった。

論理シミュレーションは大量のボタンを流す必要があり高速なシミュレーション環境が必須である。代表的なシミュレータでV810のシミュレーション速度を測定した。(図2)

この結果RT記述には内製言語である”FDL”(functional Description Language)を用い、内製シミュレータで機能検証を行なった。

V810ではテープアウト前に2週間で1200万クロックのテストボタンの正常動作を確認した。

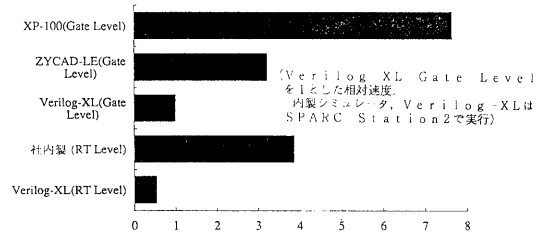


図2 シミュレーション速度

(3) 回路設計

(a) マニュアル回路設計

繰り返し構造を多用するデータバス部等は従来設計と同様にマニュアル回路設計を行った。V810のマニュアル回路設計はレイアウト設計者がレイアウトのしやすさを考慮して回路方式、回路分割等を決定した。

(b) 自動回路設計

V810は回路設計に論理合成ツールを導入した。

論理合成ツールはこれまで回路レベルの最適化機能が多く使用されてきた。V810はRTレベルからの合成を本格的に使用した。ランダム回路部分を対象にチップの約10%、2万5000トランジスタを合成した。またミスを防ぐ為合成結果の回路には一切人手による修正を加えなかった。論理合成設計の詳細は後述する。

(4) 遅延検証

過渡現象解析にSPICEシミュレータを使用し、スタティック解析は内製の遅延解析ツールを使用した。

スタティック解析は配線容量から計算した配線遅延とセルライブラリに記述されたゲート遅延から最大遅延バスを計算する。V810では制御回路と一部データバスを含む3万5000トランジスタを対象に解析し、総探索バス数は約2万5000バス、タイミング制約バス数は約3500バスである。遅延解析の詳細は後述する。

(5) ゲートシミュレーション

回路設計終了後、回路・故障検証のためゲートレベルの論理シミュレーション・故障シミュレーションを実施した。

シミュレーションはハードウェアアクセラレータを使用し、テープアウト前に57万クロックのシミュレーションを実施、RTレベルシミュレーションとの完全一致を確認した。

また、故障シミュレーションはテープアウト前に73%の故障検出率を達成し、その後テストバタンの追加により現在は93%を達成した。

(6) レイアウト設計

(a) マニュアルレイアウト

マニュアルレイアウト部は内製のポリゴンベースのレイアウトエディタを使用し集積度を上げ特性を保証した。データバス部、命令デコード部は特にチップ面積に影響するため設計途中でフロアプランを見直した。この結果、命令デコード部は設計初期に比べ面積が6%縮小した。

(b) 自動レイアウト

自動レイアウトはセルベース2層メタル対応自動配置配線ツールを使用した。レイアウト方法の詳細は後述する。

(7) レイアウト検証

レイアウト検証はすべて内製ツールで行った。DRC (Design Rule Check)、LVS (Layout Verification System)、ERC (Electric Rule Check)を実行しレイアウトの正当性を検証した。

各ブロックごとに検証を進め、最終的には1チップで検証した。従来、レイアウト検証ではツールの制約から疑似的なルール違反である「疑似エラー」が発生し、本質的なエラーを見逃すことがあったがV810はレイアウトを工夫することにより、疑似エラー箇所をすべて消去しレイアウト検証の信頼度を高めた。

6 設計データベース

設計を行っていく際にデータベースの管理は重要である。V810は各設計フェーズにおいて必要最小限のデータベースを管理することでミスを防ぎTATを改善した。

(1) RTレベル

RTレベルではRTモデルをデータベースとしRT図面、状態遷移図、仕様書等は、必要最小限のみ作成し機能の変更をRTモデルの変更のみで対応した。これによりシミュレーションモデル=仕様となるようにした。

また、RT設計を極力小人数で行いバージョン、設計者間の仕様の不整合を回避した。

(2) 回路レベル

回路図データを回路レベルの唯一のデータベースとした。内製、社外製ツールとも、この回路図データよりインターフェースを取ることができる。論理合成回路は合成ツールから直接レイアウトツール入力形式への出力が可能(EDIF等)であるがデータベースとしての意味付けのため回路図を作成した。

また従来ROM、PLA等のデータは回路図データと別形式で持つことが多いが、今回はROMコード等から自動的に回路図を作成するツールを用い他の部分と同様な回路図データを揃えた。

(3) レイアウトレベル

内製レイアウトツールのデータ形式をデータベースとする。自動レイアウト部等もすべてこのデータ形式に変換し検証を行った。

(4) ライブラリ

設計の各フェーズで使用するセルライブラリは専用のものを使用し回路シンボル、レイアウト、遅延、論理等を管理した。

7 論理合成

論理合成は現在ASIC設計に多用され、その回路品質もマニュアル設計と同等、またはそれ以上のものが作成出来るようになりつつある。論理合成ツールを使用したメリットを以下に挙げる。

(1) TATの早さ

制御回路の様なランダムロジックは設計の最後まで仕様、タイミングが決定出来ない箇所であり、TATが重視される。

(2) 論理的正確さ

ランダムロジックの設計は論理的なミスが入りやすく、また検証段階でこれを発見するのはかなり困難である。V810においてはランダムロジックをすべて論理合成で生成し、かつ集中的、網羅的な論理検証を行ない論理バグを排除した。

(3) タイミングを考慮した回路生成

ランダムロジックは熟練した回路設計者でも局所的なタイミングの最適化は可能であるが大域的な最適化は困難である。V810では当初マニュアル設計であった回路にタイミングクリティカルが存在した為修正方法の検討を行なった。この結果 ①回路の局所的修正 ②論理合成による回路レベルの最適化 ③論理合成によるRTレベルからの最適化、の3方式を評価した所③が最もタイミング的に改善され面積的にも有利(マニュアル設計と同等)であった。

尚、合成ツールを使用していくうちに次の様なデメリットがあることがわかった。

(1) 困難なマニュアルレイアウト

論理合成した回路をマニュアルレイアウトすることは難しい。回路図が自動生成の為レイアウト設計者は回路の概要がつかむことが出来ない。トランジスタ密度が要求される用途には向いていない。

(2) 回路品質評価

論理合成を行った回路は可読性が悪い為、実用に耐える回路であるかの判断が難しい。回路設計者もどんな回路が生成されているのかを完全に把握出来なくなっているのが現状である。

このため、あらかじめRT記述を工夫し誤動作する回路が生成されることを防止した。

使用出来る回路を完全同期回路、スタティック設計、セルベースの設計、等に限定した。現在の合成ツールは非同期回路には不向きでありレーシングやスパイクの問題を解決しない。また複雑なクロック方式は制御困難である。

(3) 自動レイアウトツールとの組合せ

セル数に比べ、内部ノード数が多い回路を生成した場合ではメイズルータのような配線ツールで未配線が発生する事があった。レイアウトツールに合せた回路を生成するにはある程度ノウハウが必要であった。

8 遅延解析

遅延解析は仮配線長によるフォワードアノテーションとレイアウト後に実配線長が確定した後に、行なうバックアノテーションがあり、これらの結果を設計にフィードバックした。それぞれの実行内容を挙げる。

(1) フォワードアノテーション

主として設計初期での遅延見積及びレイアウト前の論理合成による遅延制約を行なった。

マニュアル設計であるデータバス(ALU等)、端子部、ROM等の明らかなクリティカルバスをSPICEシミュレータにより解析、遅延を満足するまで回路をチューニングした。データバス部はセルベース設計以外にトランジスタレベルで最適化を行なった。

(2) バックアノテーション

主にレイアウト後、遅延制約違反箇所の発見、上位設計フェーズへのフィードバックを行なった。

サブマイクロプロセスではゲート内遅延よりも配線遅延が支配的になってくるため実配線長による遅延解析は必須である。V810においては特に制御回路とクロックドライバ部を重点的に解析した。

クロックドライバ部はクロックスキューをなくした消費電力を削減するため設計初期にチップ全体の配線長見積りから概略設計し、最終段階で再設計を行ない最適なトランジスタサイズを決定した。

タイミングの比較的明らかで無い制御部はスタティックなタイミング解析を行い動作確認をしている。

なおV810はダイナミックな遅延シミュレーションを行わなかった。これは、

(a) 完全同期回路構成を取っているためワーストケースはスタティックタイミング解析で保証出来る。ただしスタティック解析の場合False Path(起り得ないバス)の考慮が必要である。

(b) テストボタンを網羅的に作成するのは困難であり、ワーストケースがすべてチェックされているという保証が得られない。といった理由からである。

使用した遅延解析ツールは近似計算を使用しているため現在ではnsオーダーの誤差がある。このためスタティック解析でクリティカルであったバスをSPICEシミュレータで解析してワースト時間を保証する手法を取り25MHz動作することを確認した。

最終的なタイミング制約バス遅延の分布を(図3)に示す。タイミング目標を最大20ns(=

25MHz、1/2クロック)としたため20ns以上のバスが減り15~20nsのバスが相対的に増えている。

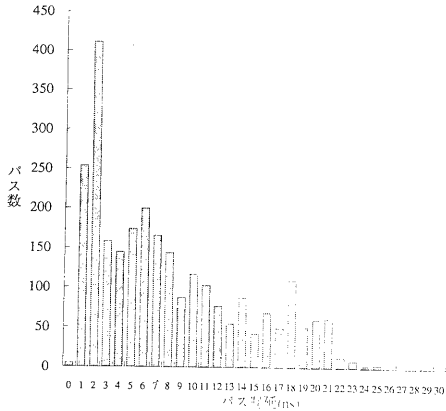


図3 バス遅延の分布

(3) 遅延解析結果フィードバック

バックアノテーションの結果タイミング制約違反箇所があった場合、この結果を上流設計フェーズにフィードバックする。V810は設計の初期段階においてはRTレベル、設計の中期以降は論理合成または自動レイアウトにおいてタイミングを改善した。ただし現在のCADツールは収束性が悪く一回のフィードバックでは必ずしも改善されないことがあった。このため最終的にフィードバックループを12回繰り返す事でタイミングを満足させた。論理合成から自動レイアウト、タイミング検証までのTATは最短で4日であった。

上記フィードバックの1手法として、レイアウトを意識した論理合成を行ないタイミングを改善した。一般に回路レベルのブロック分割はRTレベルのブロック分割がそのまま対応することが多い。しかし、クリティカルバスはブロック間配線により引き起こされることが多くタイミング改善が難しい。またレイアウトの方法として ①ブロックごとにレイアウトしてマージする ②フラットにレイアウトするといった方法が通常であるが、①の方法はブロック間の配線領域が必要となり面積の増大を招き、②の方法はクリティカルバスの配線長が保証されないため、しばしば予測と異なるタイミングになってしまう。

これを解決するためV810では論理合成と自動レイアウトツールを密に結合させる方式を使用

した。まず第一段階としてRTモデルのブロック分割とは別に、クリティカルバスが階層をまたがないよう回路のブロック分割を決定した。次にレイアウト上で各ブロックの配置位置を仮決めし、各ブロック間の距離とブロックの面積を測定し、各ブロック間配線長、ブロック内配線長の重み付けをする。一度レイアウトを行なった後は実配線長に基づき重み付けを行なう。この時点で新たに遅延解析を行ないタイミングの改善、バッファセルの挿入を合成ツールに行なわせる。自動レイアウトでは(図4)の制御部の様に全体を一つの領域にレイアウトし、さらにその中でブロックごとの配置領域を強制的に指定する方法をとる。これにより (a) ブロック間配線長をより確実に予想出来る (b) 配線専用の領域が無くなる (c) ブロック内の総配線長の短縮 (d) データバスと制御部の位置を合わせることでジョグ(配線の曲り)を無くせるため面積が縮小できる、といった効果があった。

この結果、設計当初より自動レイアウト部面積は22%減少し、最悪クリティカルバスは31%改善した。

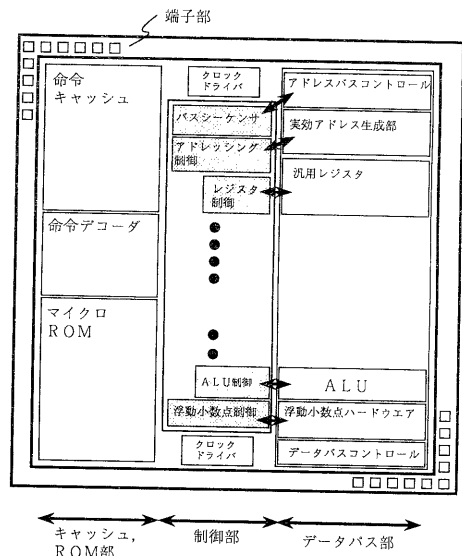


図4 チップフロアプラン

9 消費電力低減

回路の消費電力は一般に次の要因から決定される

[文献1]。

- (1) 電源電圧
- (2) 各ノードの容量
- (3) 各ゲートの0/1遷移頻度

マニュアル回路においては上記(2)(3)を考慮し、回路構成を工夫することで、電力消費を低減した。

V810の自動設計手法を使用した回路においては、主に(2)を低減することで低消費電力を実現した。この方法を以下に示す。

- (1) タイミングクリティカルでない箇所についてディメンジョンの小さいゲートを優先的に使用した。
- (2) 内部ノード数を少なくするため冗長セルを減らし、また複合ゲートを優先的に使用した。
- (3) セルのファンアウト数と平均配線長は比較的比例する。ただし自動レイアウト部ではあるファンアウト数以上になると配線容量が飛躍的に増大し始めることが分かった。そのため最大のファンアウト数を制限した回路を生成し、総配線長を短縮した。
- (4) バックアノテーションの項で述べたレイアウト時の配置制約によって、狭い領域にレイアウトし、それぞれの配線の長さを短縮した。

10 おわりに

V810の設計開始からテーブアウトまでに要した期間は一年未満であった。

以下にチップの評価結果を示す。

(1) 機能評価

ファーストシリコンにおいて全ての命令機能が正常に動作し、加えてリアルタイムOS、SPECベンチマークなどを含む幾つかのアプリケーションが動作した。なお、2万桁の円周率(π)を490.4秒で計算出来ることも確認した。

(2) 特性評価

ファーストシリコンは室温で電源電圧2.2V時に20MHzで動作した。

実チップ内のクリティカルパススピードを確認するため、遅延解析でクリティカルであるとされた遅延パスを再現し実測した。(図5)実チップはSPICEシミュレーションに比べ最大で20%早く動作した。この誤差はレイアウトからの配

線容量見積の誤差、測定誤差が大部分であると考えられる。現在V810の最高動作周波数25MHzであるため、この程度の誤差は許容範囲ではある。しかし今後、高MHzでの設計ではさらに一桁以上精密な遅延解析が必須である。

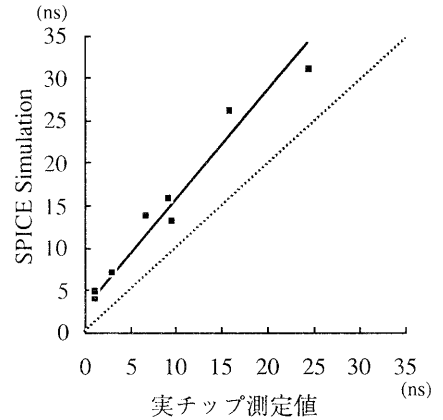


図5 実チップ遅延とシミュレーションの相関

11 参考文献

- [1] 菅野監修、飯塚編「CMOS超LSIの設計」、培風館、1989年
- [2] 楠田、平位他「低消費電力・低電圧動作のオリジナル32ビットRISCマイクロプロセッサV810」、NEC技報 V0145, No. 8、1992年
- [3] 針谷、楠田他「低消費電力・低電圧動作の32ビットマイクロプロセッサ V810」、情報処理学会研究報告 92-ARC-96、1992年
- [4] 山畑、黒田他「2.2V動作を可能にしたV810」、日経エレクトロニクス、1992年 11-23号