

フルカスタム・マイクロプロセッサの設計手法最適化

宇佐美 公良 岩村 淳

(株)東芝 半導体技術研究所

はじめに フルカスタム・マイクロプロセッサの開発において、高性能でできるだけ面積の小さなチップを作り上げることが重要であることは言うまでもないが、「いかに開発期間を短くするか」というファクタが、近年ますます重要になってきている。

今回我々は、32ビットマイクロプロセッサTX1の開発にあたり、「短期間で設計すること」を主眼に置き、その中でチップサイズを極力抑え高性能を保つ方針の下で、設計手法の最適化を図った[1]。本報告では、TX1の設計手法を紹介するとともに、チップサイズ、設計工数及び設計期間の観点から本手法を評価する。

TX1の概要 TX1は、TRONアーキテクチャ[2]に基づくCISC型マイクロプロセッサである。2層メタルの1 μ m CMOSプロセスを用い、460Kトランジスタを10.89 \times 10.27mm²のチップ上に実現している。

TX1の設計手法 TX1の設計手法の特徴は、制御ロジックは自動論理合成及び自動レイアウトによる自動設計を行い、いっぽうデータバスは、マニュアルで設計したことである。図1に、設計フローを示す。

まず機能設計では、チップ仕様書をもとに機能モジュールに分け、H²DL[3]と呼ばれるレジスタトランスファ・レベルの記述言語を用いて、機能を記述する。ここで重要な点は、H²DL記述を行う際に、①制御ロジック②データバス③マクロセル (ROM, RAM, PLA) の3つのカテゴリーに分類する事を意識して、機能モジュールをさらに細かいブロックに分割したことである。このあとの論理設計・レイアウト

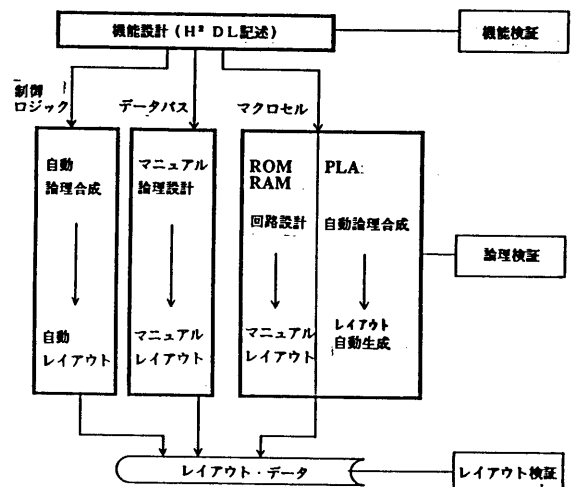


図1. 設計フロー

設計の手法は、カテゴリーによってそれぞれ異なる。制御ロジックでは、H²DL記述をもとに自動論理合成[4]を行い、得られたネットリストから、スタンダードセルを用いた自動レイアウトを行った。いっぽうデータバスは、マニュアルで論理設計とレイアウト設計を行った。また、マクロセルの中でも、ROM, RAMはマニュアル設計し、PLAは論理合成・真理値表の最小化およびモジュールジェネレーティングにより、完全自動設計した。図2はTX1のチップ写真であり、図3は各ブロックと上記3つのカテゴリーとの対応を示している。

このように、制御ロジックとデータバスは、全く異なる手法で設計したが、これは次のような理由に基づいている。

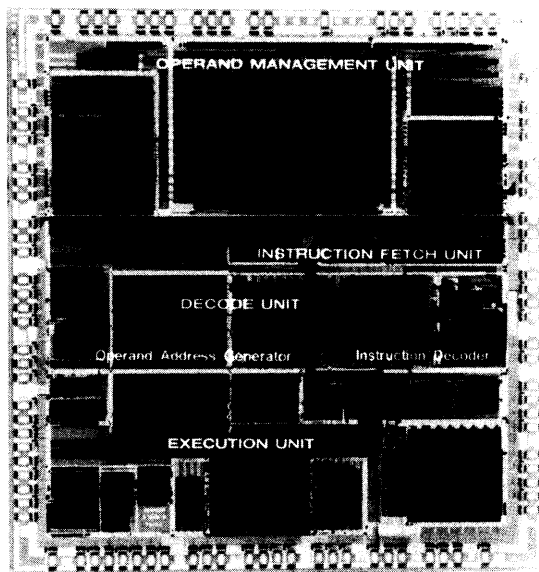
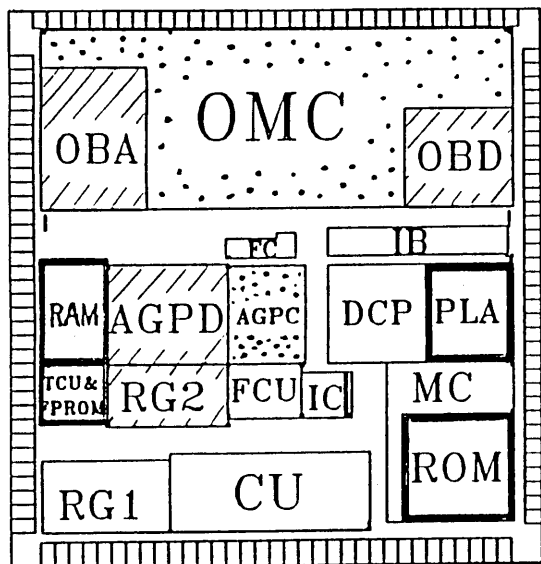


図2 TX1 チップ



- 制御ロジック
- データバス
- 制御ロジック・データバス (mix)
- マクロセル

図3 各ブロックの性質

データバスは、ビットスライスの規則的な構造をしているため、マニュアル設計でも生産性を落とすことなく、むしろ高密度で高性能のものが実現できる。これに対し、複雑なステートマシンを持つ制御ロジックを、マニュアルで論理設計しレイアウトした場合、論理が複雑で規則性が低いいため、設計作業はもとより、デバッグ作業に長大な時間がかかってしまう。この点を考慮すると、制御ロジックは機能レベルで記述し、論理設計及びレイアウトを自動で行うのが効率的である。

デザイン・パイプライン 我々はさらに、各設計工程を効率的にオーバーラップさせ、設計期間全体の短縮化を図る「デザイン・パイプライン」という新しい手法を試みた。従来の設計手法では、図4に示すように、機能設計・機能検証が完了してから論理設計を開始し、また論理検証が完了してからレイアウト設計を開始した。このようなシーケンシャルな進め方でTX1を設計した場合、設計期間は27.5ヶ月（約2.3年）かかるものと推測される。TX1の各設計ステージに要した時間を表1に示す。

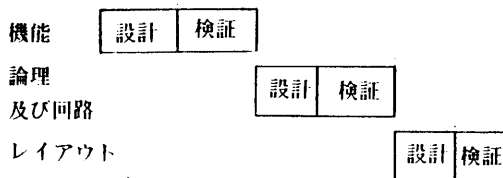


図4. 従来の設計手法

表1. 各設計ステージに要した時間

設計ステージ	時間 (月)
機能	12.0
論理及び回路	9.0
レイアウト	6.5
合計	27.5

さらに詳しく分析すると(図5)、設計作業そのものにかかる時間よりも検証にかかる時間のほうが数倍も大きく、機能検証は全設計時間の1/3、また論理検証は全設計時間の1/4もの時間を占めている。機能検証にこれだけの時間がかかったのは、機能シミュレーションと並行して、テストベクタの作成を行ったことが大きく影響している。

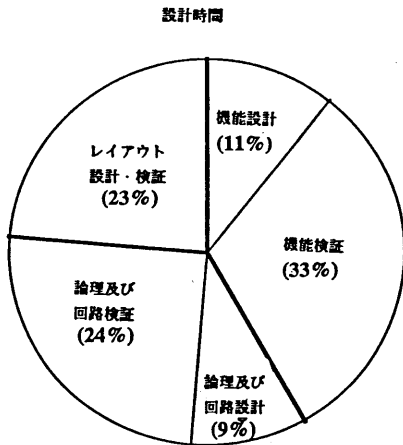


図5. 各設計ステージに要した時間の比率

以上の分析から、上位設計工程の検証が完全に終了するまで下位の設計工程が待たされるという進め方では、設計期間を短縮することは困難と判断されたため、我々は図6に示す「デザイン・パイプラインニング」という新しい手法を導入した。すなわち、機能設計(H²DLでの記述作業)を終え、基本的な機能動作の確認を終えると、チップ全体の詳細機能検証と並行して論理設計を開始した。論理設計は、マニュアルで設計するデータベースのほうを先に着手し、自動論理合成する制御ロジックは少し遅れて着手した。また、レイアウト設計も同様にチップ全体の論理検証と並行して進めた。以上のようなデザイン・パイプラインニングにより、全設計期間を12.5ヶ月に短縮することができた。

ところで、デザイン・パイプラインニングは、図1に示す設計手法と密接な関係がある。デザインパイプラインニングでは、機能検証を行ないながら論理・レイアウト設計を並行して走らせるので、もし機能検証で

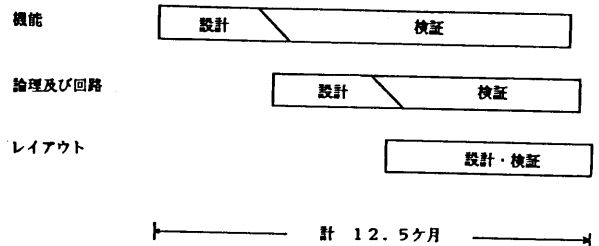


図6. デザイン・パイプラインニング

バグが見つかった場合、下位工程である論理・レイアウトにも修正が入るわけで、この修正のターンアラウンドが重要なカギとなる。図1の設計手法では、データベースをマニュアル設計し制御ロジックを自動設計するが、

- 1)機能検証でバグが発見され修正が入る頻度は、制御ロジックで高く、データベースでは低い
- 2)制御ロジックは、論理設計及びレイアウトを自動で行なっているため、修正に要する時間は短い

という理由から、修正が入った場合でも、デザインパイプラインニングは、設計期間全体を短縮する効果を発揮できる。

設計手法の評価

TX1の設計手法の評価にあたって、制御ロジックのみならずデータベースも自動設計する完全自動設計手法と比較した。そして、TX1に完全自動設計手法を適用した場合、チップサイズ・設計工数がどうなるかを見積もった。なお、マクロセルについては、完全自動設計手法の場合もTX1の手法と同様の設計手法を採るものとした。

(1) チップサイズ

完全自動設計手法ではデータベースが、スタンダードセルを用いた自動レイアウトの平均素子密度(1.6K素子/mm²)でレイアウトされると仮定し、面積を見積もった。結果を表2に示す。自動レイアウトの素子密度は、マニュアル設計したデータベースのそれに比べ27%と低いため、データベース部面積の増大を招き、結果的に完全自動設計手法は

TX1の設計手法に比べ、チップ面積が70%大きくなるのがわかる。

表2. チップ面積の比較

	本手法 (mm ²)	完全自動設計 (mm ²)	面積増加 (mm ²)
制御ロジック	23.7	23.7	0
データベース	17.2	62.0 *	+44.8
制御ロジック・ データベース	20.4	54.5 *	+34.1
(マクロセル	11.1	11.1	—)
I/O	39.4	39.4	—)
合計	111.8 (10.6mm ²)	190.7 (13.8mm ²)	+78.9 (+70%)

* 素子密度=自動レイアウトの平均密度と仮定し算出

(2) 設計工数

完全自動設計手法での設計工数を調べるにあたり、当研究所で開発した画像処理LSI [5] を例にとり、工数分析を行なった。そして、TX1に完全自動設計を適用した場合の工数を見積もったのが、図7である。

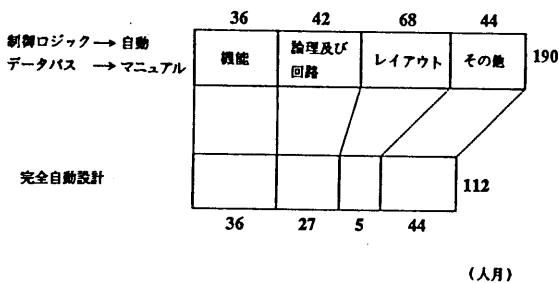


図7. TX1を完全自動設計した場合の設計工数(見積り)

データベース部にも自動設計を適用する完全自動設計手法では、TX1の手法に比べ、論理設計工数を64%に(42人月→27人月)、またレイアウト工数を5%(68人月→5人月)に減らすことが可能と予測される。結果として、TX1の設計手法で190人月かかっていた全設計工数が、完全自動設計では112人月に減少すると見積もられる。

結論 フルカスタム・マイクロプロセッサの設計手法として、制御ロジックには、自動論理合成及びスタンダードセル方式の自動レイアウトを徹底的に活用する自動設計を行い、いっぽうデータベースはマニュアルで論理設計及びレイアウトを行う手法を提案した。さらに、上記手法の利点を活かしながら、各設計工程を効率的にオーバーラップさせて進めていく「デザイン・パイプライニング」という手法を、新たに試みた。

以上述べた設計手法を、32ビット・マイクロプロセッサTX1に適用した結果、チップ設計開始から動作チップを得るまで、約1年という期間で設計できた。この設計期間短縮化には、デザインパイプライニングが大きく寄与しており、デザインパイプライニングの導入なしでは2、3年の設計期間を要するものと予測される。また、制御ロジックのみならずデータベースにも自動設計を適用する完全自動設計手法と、本設計手法とを比較した結果、本手法は1.7倍の設計工数(人月)がかかるものの、チップサイズは完全自動設計の場合に比べ、60%に抑えられることが見積もられた。以上の評価結果から、開発期間が短縮でき、かつチップサイズを極力押さえ高性能のチップを得るための設計手法として、本設計手法が有効であることがわかった。

参考文献

- [1] K. Usami and J. Iwamura, "Optimized Design Method for Full-Custom Microprocessors", Proceedings of the CICC '89, May 15-18, 1989.
- [2] K. Sakamura, "The TRON Project", IEEE Micro, vol. 7, no. 2, Apr. 1987, PP. 8-14.
- [3] 西尾他, "階層的ハードウェア設計言語H²DL", 昭60信学全大 S9-2.
- [4] 増淵他, "自動論理合成システムLUNAを用いた論理回路設計", 第33回(昭61)情処全, R-6.
- [5] N. Kai et al., "A High Speed Outline Font Rasterizing LSI", Proceeding of the CICC'89, May 1989.