

A I チップ (I P 1 7 0 4) の開発環境

2W-2

岡村 光善* 木南 英志** 今井 徹* 相川 健* 斎藤 光男*
 * (株) 東芝 総合研究所 **東芝ソフトウェアエンジニアリング (株)

1. まえがき

VLSIプロセッサを開発する際には、アーキテクチャ設計、論理設計、レイアウト設計、タイミング検証、実チップ評価の段階がある。ASICの設計に比較するとアーキテクチャ設計に重点がおかれ、この段階において十分に仕様、性能の評価を行なう必要がある。近年ASICをターゲットとするVLSI・CADについては多くの優れたシステムが開発されているが、VLSIプロセッサの開発を行なう場合にはアーキテクチャ設計に対して十分な機能を持っているとはいえない。

VLSIプロセッサ開発効率の向上の為に、①上記の各段階、特にアーキテクチャ設計の段階で十分なデバッグ、検証手段を有すること、②検証に使用されるテストプログラムが共有できること、③できる限りマンマシンインタフェースが統一化されていること、が重要である。

我々はAIチップ(IP1704)の開発にあたり、上記の考え方に基きアーキテクチャ設計から実チップ評価に渡る各開発行程をサポートするツールの開発を行った。このツールを使用することにより、AIチップを効率的に開発することが可能であった。

2. 全体構成

AIチップ開発環境は、命令仕様定義の段階から、実チップ評価の段階までをサポートするツール群であるが、大きく分けて次の4つの部分からなる。

- ①命令仕様定義、マイクロプログラム作成ツール
- ②テストプログラム作成ツール
- ③アーキテクチャ設計、論理設計、レイアウト設計の検証のためのシミュレーションモデル
- ④実チップ、周辺ハードウェアを実装した評価用ハードウェア

図1にAIチップ開発環境の全体構成を示す。

3. 命令仕様定義、マイクロプログラム作成ツール

命令フォーマット、命令の動作、アセンブラフォーマット、マイクロプログラム等が記述されている命令仕様書ファイル、及び命令コード割付けファイルの情報を基にマイクロコードを生成するツールを開発した。このツールで作成されたマイクロコードは、アーキテクチャ検証

用シミュレータ、論理検証用シミュレータで共通に使用可能である。なお、4.3で述べるアセンブラ、逆アセンブラは、命令仕様書ファイルを変更すると自動的に更新される。

4. テストプログラム作成ツール

アセンブラ、リンカ、コンパイラの開発を行なった。これによりアセンブラ及び高級言語でテストプログラムを作成できる。このテストプログラム作成ツールで作成されたテストプログラムはアーキテクチャ検証、論理設計の検証、実チップ評価で共通に使用できる。

5. ハードウェア・アーキテクチャ検証ツール

アーキテクチャ検証の段階ではアーキテクチャの正統性と性能評価の検証が必要である。この2つの検証を行なうためにAIチップの論理シミュレータ、及び論理シミュレータをコントロールするためのデバッグの開発を行なった。

5.1 論理シミュレータ

論理シミュレータはプロセッサとメモリのモデルをレジスタトランスファーレベルで記述したものである。AIチップの5段のパイプラインステージ(フェッチ、デコード、実行、メモリアクセス、ライトバック)のモデルがインプリメントされており、ディレイドランチ、スクワッシュランチも正確にシミュレートされる。記述言語としてはデバッグ手段としてシンボリックデバッグが使用できることからCを用いた。AS3000上でシミュレータのプログラム実行速度は1クロック当たり0.3~1msであった。

5.2 デバッグ

デバッグは上記の論理シミュレータをコントロールするものであるが以下の機能を持つ。

①ハードウェアリソース・アクセス機能

メモリ、AIチップ内のハードウェアリソースにアクセスする機能。

②プログラム実行機能

テストプログラム作成ツールで作成されたテストプログラムを実行する機能。

その他、マイクロ及び機械語レベルの1ステップ実行、ブレークポイント設定、等の機能。

③I/O実行機能

AIチップ上で走るテストプログラムのI/Oを肩代わりする機能。

Environment for Developing an AI Processor

Mitsuyoshi OKAMURA*, Hideshi KIMINAMI**, Toru IMAI*
 Takeshi AIKAWA*, Mitsuo SAITO*

* TOSHIBA Corporation. **TOSHIBA Software Engineering

5.3 検証

上記のシミュレータ上で、デバッガを用いてC、P R O L O G、L I S P で書かれたベンチマークプログラムを実行し、アーキテクチャの正統性と実行クロック数の評価を行なった。

6. 論理設計・レイアウト・サポートツール

この段階では、アーキテクチャ設計をいかに忠実にインプリメントするかがキーポイントとなり、検証もアーキテクチャ検証時に用いたテストプログラムを走らせた結果が同一の動作を行なうことを確認する必要がある。この段階でのツールとしては、シリコンコンパイラ *genesil* を使用した。

デバッガは、*genie* と呼ばれる L I S P ライクの言語を用いて、5章で述べたデバッガとほぼ同一のマンマシンインタフェースを持つものを開発した。シミュレーションモデルは *genesil* により論理設計データから自動的に作成される。テストプログラム作成ツールで作成されたテストプログラムはこの環境でも使用できる。

シミュレーションは A I チップとキャッシュのモデルを用い、ブロックレベル、ゲートレベルの2段階で行なった。

7. 実チップテスト環境

実チップテストの段階では、5章で述べた論理シミュレーション、及び、6章で述べた機能シミュレーションで検証したことが、実チップで正常に動作するかのデバッグ、検証が必要である。

7.1 デバッグ

実チップテストのためのデバッガは、論理シミュレーションで使用したデバッガを利用し、下位レベルのドライバルーチンのみを変更したものであるため、マンマシン・インタフェースは全く同じである。さらに、アーキテクチャレベル、論理設計レベルのシミュレーションで用いたテストプログラムもそのまま使用できる。

7.2 評価ボード

評価ボードの基本構成は、図2に示すように、データメモリ、タグメモリ、スキャンバス制御、割込み制御、トラップ制御等から構成されており、VMEインタフェースを介してデバッガから制御される。また、実チップ評価には機能動作確認の他に、動作速度の評価が重要である。そこで、評価ボードはチップの基本的な機能はハードウェアで処理し、リアルタイムでの動作を可能にしている。キャッシュメモリのコントロールロジックは A S 3 0 0 0 上のソフトウェアでシミュレートしている。

8. おわりに

命令仕様定義、アーキテクチャ設計から実チップ評価までをサポートするツールを開発した。このように設計の各段階で計算機サポートされたツールを用い検証を行なうことにより、確実な設計が可能となりかつ設計のターンアラウンドタイムが短縮される。また実際にこれらのツールを使用してみて、各設計段階において使用するデバッガのマンマシンインタフェースを統一する事は、使い勝手を非常に向上させる事がわかった。

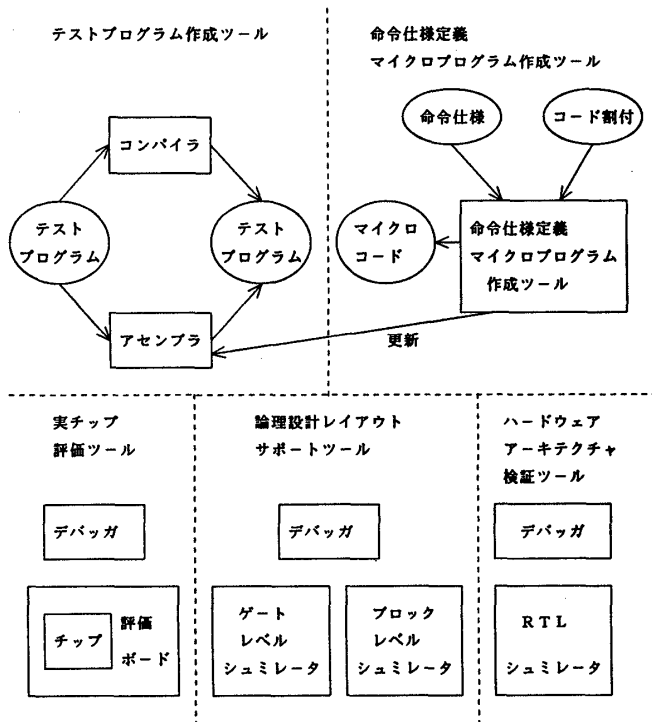


図1 AIチップ開発環境の全体構成

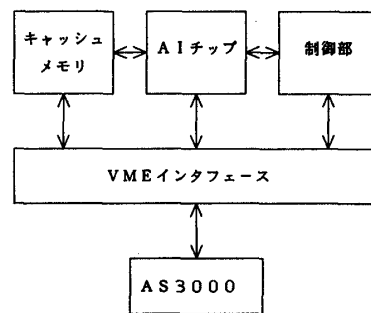


図2 評価ボードの構成