

2U-7

TRON仕様32ビットマイクロプロセッサGMICRO/100

(3) 関係データベースを中心に統合されたマイクロプログラム開発ツール

澤井克典 清水徹 岩田俊一 吉田豊彦 日向純一 富沢治

(三菱電機株式会社 LSI研究所)

1. はじめに

マイクロプログラム制御方式は高機能、高性能なマイクロプロセッサを実現する上で重要な方式の一つである。マイクロプロセッサ内部の機能ブロックを並列に制御することにより、従来ソフトウェアで記述していた機能を効率よく実現することができる。

しかし、VLSIの集積度の向上にともないマイクロプログラムで制御すべき機能ブロックは増大しており、これらを効率よく動作させて高機能を実現するためにはマイクロプログラムのアルゴリズムが複雑になる。このようなマイクロプログラムを短期間に開発するため、マイクロプログラム開発ツールが重要となる。

本稿ではTRON仕様に基づいた32ビットマイクロプロセッサGMICRO/100のマイクロプログラムを効率よく開発するために作成した開発ツールについて述べる。

2. マイクロプログラム開発ツールの設計方針

マイクロプログラム開発には、主に次のような作業が含まれる。

- マイクロアーキテクチャの設計
- マイクロプログラムの設計
- ROM化

マイクロアーキテクチャの設計では、マイクロ命令のフィールド構成と各フィールドで指定できるマイクロオペレーションを決定する。マイクロアーキテクチャが決まると、マイクロプログラムの設計が行われる。マイクロプログラムをコーディングし、シミュレーションに基づいてデバッグする。また、マイクロプログラムのアドレスを決定し、最適化を行う。最後に、マイクロプログラムを実際のチップ上にマイクロROM化する。

このようなマイクロプログラム開発の各フェーズに対応するため、次のような方針をとった。

- マイクロプログラムを関係データベースで管理する。
- 用途別にツールを用意し、一つのマイクロプログラムコードから各種のデータを生成する。
- 一つのマシン上にツールを統合し、ツール同士を組み合わせて使えるようにする。

これにより、マイクロプログラムに関するデータの多重化やデータの不整合をなくすこと、種々の仕様変更に対して柔軟かつ速やかに対処すること、マイクロプログラム開発に用いるツール群を短期間に開発することを狙った。

3. GMICRO/100マイクロプログラム開発ツール

図1にGMICRO/100のマイクロプログラム開発ツールの構成を示す。図中、マイクロプログラム・ソースファイルは、マイクロプログラムのアルゴリズムを記述したものである。

以下、開発環境を構成するツール群を順に説明する。

① マイクロプログラム・アセンブラ

マイクロプログラムのアルゴリズムが記述されているマイクロプログラム・ソースファイルをデータベースのフォーマットに対応したコードに変換するツールである。データベース・フォーマットは、マイクロ命令のフィールドに対応したフィールドに分けられており、各フィールドにはマイクロプログラムが指定するオペレーションがニモニックで記述されている。

② マイクロプログラム・逆アセンブラ

データベース・フォーマットで記述されているマイクロプログラムコードを、アルゴリズム記述のフォーマットに変換する。マイクロプログラムのレビューやドキュメントの作成に用いる。

③ データフロー・アナライザ

データベース・フォーマットのマイクロプログラムコードから、ワーキングレジスタの使用状況を調べるツールである。

GMICRO/100は、ストリング操作命令、可変長ビットフィールド操作命令等の高機能命令を実装する。これらの命令を実行するマイクロプログラムは、一時的に保持すべきデータ数が多く、頻繁にワーキングレジスタを使用する。制御の流れも複雑であり、人手でワーキングレジスタの使用状況を調べ、必要最小限のワーキングレジスタの数を割り出すのは困難である。今回作成したデータフローアナライザは、制御の流れに沿ってワーキングレジスタの使用状況を調べ、マイクロプログラムのどのワードで、どのワーキングレジスタが使用されているかを表示するものである。これによって、ワーキングレジスタの使用状況が一目でわかり、人手によるワーキングレジスタの割り付けが可能となる。このツールを用いてワーキングレジスタの数を最適化した。

④ 機能シミュレータ用フォーマット変換ツール

データベース・フォーマットのマイクロプログラムコードを、GMICRO/100の機能シミュレータが受け付ける入力形式に変換する。機能シミュレータを用いてマイクロプログラムのアルゴリズム検証を行った。

Implementation of a 32-bit microprocessor GMICRO/100 based on TRON specification (3)Microprogramming support system on the relational database

Katsunori SAWAI, Toru SHIMIZU, Shunichi IWATA, Toyohiko YOSHIDA, Junichi HINATA, Osamu TOMISAWA
MITSUBISHI ELECTRIC CORPORATION

⑤ ビットパターン・ジェネレータ

データベース・フォーマットのマイクロプログラムコード（ニモニックで表されている）を、2進数または16進数のビットパターンに変換する。このビットパターンを論理シミュレータに入力し、論理シミュレーションを行った。また、マイクロROMの設計にも用いた。

⑥ データベース

データベース管理システムにはINGRESを使用した。マイクロプログラム・アセンブラで生成したデータベースを検索することにより、マイクロプログラムの保守、マイクロアーキテクチャ設計、ハードウェア設計に使用した。

以上の開発支援ツールは、主にプログラミング言語Cで記述され、UNIX上で動作する。

4. 評価

表1にマイクロプログラム開発ツールのプログラムサイズ及び使用したプログラム言語を示す。

表1 マイクロプログラム開発ツールのプログラムサイズと言語

開発ツール	ステップ	言語
マイクロプログラム・アセンブラ	2800	C、AWK
データフロー・アナライザ	1000	C
マイクロプログラム・逆アセンブラ	600	AWK
ビットパターン・ジェネレータ	2500	C
機能シミュレータ用フォーマット変換ツール	500	AWK

表に示すように、各ツールのプログラムサイズは小さい。また、どのツールも1~2週間という短期間で開発することができた。これは、各ツールが同一のデータベース・フォーマットのマイクロプログラムコードを参照するため、一度開発したツールのコードを再利用できることと、用途別にプログラムを作成したことによる。

5. 結論

関係データベースを中心に統合したG_{MICRO}/100マイクロプログラム開発ツールを構築した。関係データベースは、マイクロアーキテクチャ設計、ハードウェア設計に用いた。開発支援ツールとして、マイクロプログラムアセンブラ、逆アセンブラ、データフローアナライザ、ビットパターンジェネレータ、機能シミュレータ用変換ツールを開発した。これらのツールは、同一のソースファイルを参照することで、データの不整合が生じないようにした。

今後、これらのツールをさらに整備することにより、次機種展開でのマイクロプログラム開発に適用していきたい。

謝辞

本研究に関し、有意義な討議と助言を頂いた東京大学坂村助教授をはじめ、G_{MICRO}/100の開発に携わった方々に感謝致します。

<参考文献>

- 1) 相磯、飯塚、坂村:「ダイナミックアーキテクチャ」共立出版、(1980)
- 2) 「マイクロプログラミング技術」、情報処理、vol.28(1980),No.12

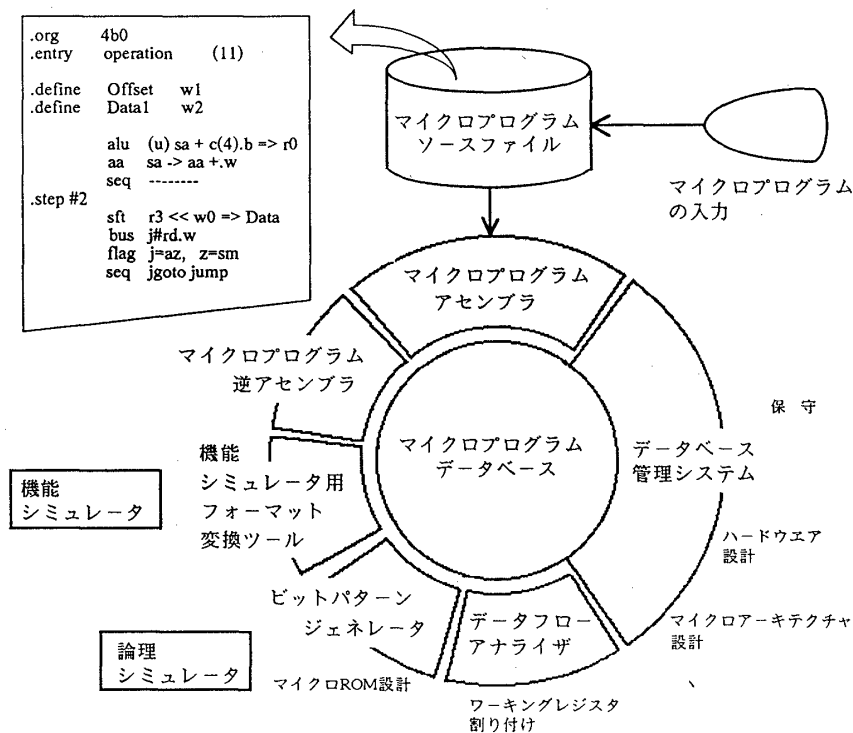


図1 マイクロプログラム開発ツールの構成