

2S-2

計算機ハードウェア教育のための支援システムの開発

古田勇次* 岡田光博* 井上訓行**

furuta@ksuics.kyoto-su.ac.jp

京都産業大学 *理学部 **工学部

1. はじめに

ハードウェア教育の環境を整備するため、計算機の内部観測、論理設計、レイアウト、マイクロプログラミング、機械語プログラミングなどの広範囲の教育を支援する環境を構築した。特に教育用計算機はほとんど全部書換可能デバイスで実現されているので、各種の要求に柔軟に対応できる。また、LSI化された計算機の内部観測支援にはパソコンを利用して、デバッグ環境のユーザーインタフェースを飛躍的に向上させている。

2. 背景と目的

集積回路技術やCAD技術の発達により、大学でのLSI設計教育が益々重要になってきている。ソフトウェアの教育環境はいろいろ開発されているが、大学でのハードウェアの教育環境は著しく立遅れている。ソフトウェア開発環境のような使いやすさで、LSIの設計から試作まで初級者でも容易に行える支援システムが必要となっている。一方、学生実験の時間的、予算的制約からチップの製作を外部に依頼したり、必要なだけICテストやロジックアナライザなどの測定器を用意することはできない。

最近教育用プロセッサチップをASIC技術やFPGAを用いて開発した事例は報告されている¹⁾⁴⁾。我々の大学でも数年前からLSI教育の実習においてCADツールを用いた回路図入力と論理シミュレーションまでは行っている。しかし、実際に動作する計算機の製作にはTTLなどのゲートレベルICを使用している。このため設計そのものより半田付けやラッピング配線などに時間を費やす場合が多い。このような背景から、研究室内で学生自身が独自の計算機を設計したり容易に仕様を変更でき、かつ設計した計算機を短期間にLSI化し動作確認まで行えるような計算機ハードウェア教育のための支援環境を構築した。

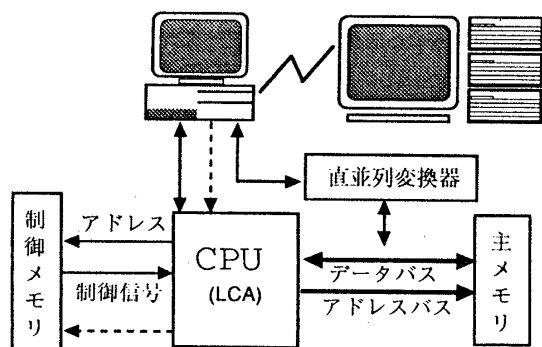


図1 ハードウェア構成

3. システム構成

本システムのハードウェア構成を図1に示す。教育用計算機をLSI化するためのデバイスには、製作の時間と費用(書換の容易性)の点からXILINX社のFPGA(Field Programmable Gate Array)であるLCA(Logic Cell Array)を採用した。LCAはスタティックRAM方式のFPGAで書き込み回数に制限がなく、構成データをロードするだけで直ちにチップが完成する。また、設計、変更、デバッグの容易性を重視してマイクロプログラム制御方式を提供することにした。制御メモリを使用せずに結線制御も可能である。制御メモリと主メモリには、書換の容易性を優先してスタティックRAMを使用しており必要に応じてピン互換のROMと交換できる。教育用計算機を実現するためのボードには上記のLCA、制御メモリ、主メモリの他に、LCAの構成データをダウンロードするためのROM、クロック生成回路、切替スイッチなどを搭載している。コネクタ部へはパソコンとのデータ転送をするために必要な配線が施されている(図2)。

本システムでは、パソコンは次の4つの目的に使われている。また、ネットワークを通してワークステーションに接続されている。

- ①LCA構成データの作成(配置配線)
- ②LCA構成データの供給(プリンタポート)
- ③計算機の入出力装置(シリアルポート)
- ④デバッグ支援ツール(パラレルボード)

表1に本システムに使用するソフトウェアを示す。回路図エディタ、論理シミュレータはワークステーション上の汎用CADシステムを使用している。LCAへのテクノロジーマッピングにはパソコン上のXACTを用いる。パソコンが入力出力装置として動作するときにはターミナルエミュレータを使用する。以上は市販のソフトウェアである。マイクロプログラムの作成には、パソコン上のマイクロアセンブラを用いてレジスタトランスファ記述から、2進形式のマイクロ命令を生成する。機械語のプログラムもパソコン上のクロスアセンブラでアセンブルして、主メモリにダウンロードする。

表1 ソフトウェア構成

ソフトウェア	名称	計算機
回路図エディタ	HP-DCS	ワークステーション
論理シミュレータ	System HILO	〃
配置配線プログラム	XACT	パソコン
ターミナルエミュレータ	VT282Iエミュレータ	〃
デバッグ支援ツール		〃
マイクロアセンブラ		〃
クロスアセンブラ		〃

4. 教育用計算機的设计手順

図1のシステムを用いてマイクロプログラム制御方式の計算機を設計する手順の概略を図3に示す。結線制御方式の場合もほとんど同じである。

- ①計算機の仕様を決める。ここではアーキテクチャ、命令の形式、タイミングを決める。命令セットを決める必要はない。このとき、マイクロプログラムをダウンロードするために、データバスから制御メモリへの転送経路と、転送制御ロジックを組込んでおく。
- ②設計した回路を回路図エディタを用いて回路図または論理式で入力し、ネットリストを生成し、論理シミュレーションを行う。この部分は汎用のCADシステムを利用している。
- ③設計した論理をLCAへマッピングするため、ネットリストを変換する。この変換はロジックをLCAのブロックにマッピングし、ブロック間の接続情報を生成する。つぎに各ブロックをLSI上の特定の位置に配置し、そのブロック間を配線する。配置配線は自動でも手動でも可能である。この部分はテクノロジーに依存するためパソコン上で専用ソフトXACTを使用する。
- ④配置配線の結果得られた構成データをLCAにダウンロードする。また構成を固定するときはROMライターへ送りROM化する。ダウンロードする機能はパソコンのソフトと、LCAに備わっている。LCAのこの部分が本計算機でただ1つRAMでない部分である。(チップ完成)
- ⑤命令セットを決め、マイクロプログラムを作る。マイクロプログラムをレジスタトランスファ形式で書き、マイクロアセンブラによって2進形式に変換する。このとき、機械語のプログラムを主メモリにダウンロードするマイクロプログラムを入れておく。
- ⑥マイクロプログラムをダウンロードするロジックを起動して、マイクロプログラムを制御メモリに移す。(プロセッサ完成)
- ⑦機械語のプログラムを書き、クロスアセンブラで2進形式に変換する。
- ⑧ダウンロード用のマイクロプログラムを起動して機械語のプログラムを主メモリにダウンロードする。

5. デバッグ支援

作成したチップのデバッグにはマイクロ命令の供給、レジスタへのデータ設定、内部の観測表示が必要である。このためにチップ内に観測専用のバスを設ける方式¹⁾や必要な信号線を全てチップの外に出す方式²⁾がとられる。本システムではマイクロプログラム制御方式の特徴を利用して次の方式をとる。

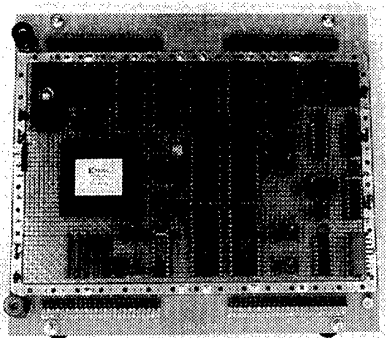


図2 教育用計算機ボード

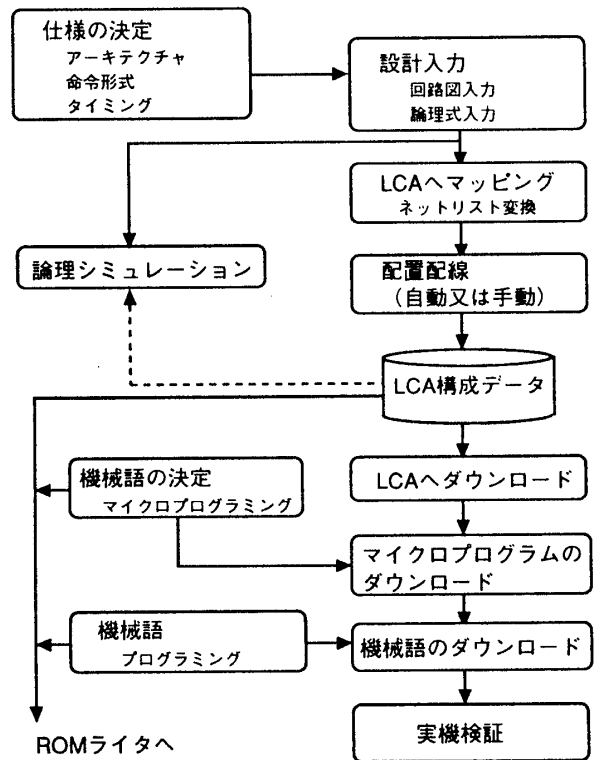


図3 プロセッサ設計手順

レジスタトランスファ文を入力すると、パソコンがこれを解析しマイクロ命令を生成する。また、パソコンからバス上にデータを設定できる。データバスに必要なデータを出しておき、必要なマイクロ命令を生成して、クロックを1つ進めることによって、任意のデータを指定したレジスタにセットすることができる。

一方、プロセッサの内部状態を観測するために、主メモリとI/Oリードの時以外は内部で転送されるデータが外部データバスへ出力されるように設計しておく。マイクロ命令とクロックを観測し、適当なタイミングでバス上のデータをパソコンに取込んで表示をする。観測が間接的になるが、チップ内に追加部分を必要としないでレジスタの内容を観測することができる。

6. おわりに

16ビットと8ビットのプロセッサを試作することによって、本システムの有効性を確認した。マイクロアセンブラは設計したプロセッサごとに作り直さなければならないが、現在制御ビットとレジスタトランスファ記述の対応関係を定義したファイルを用意することにより2進形式に変換できる汎用性のあるマイクロアセンブラを開発中である。また、パソコンによるデバッグ支援ソフトのユーザーインターフェースをより一層向上させるため、マルチウィンドウ対応版の作成を計画している。

参考文献

- 1) 神原弘之: 教育用マイクロプロセッサ: KUE: CHIP, 情報処理学会計算機アーキテクチャ研究会, 82-6 (1990)。
- 2) 神原弘之他: 教育用マイクロプロセッサ: KUE: CHIPの開発と教育への応用, 情報処理学会第43回全国大会, 2Q-13 (1991)。
- 3) 末吉敏則他: 書換可能なLSIを用いた教育用マイクロプロセッサの開発, 情報処理学会第43回全国大会, 2Q-11 (1991)。
- 4) 田中康一郎他: 書換可能なLSIを用いた教育用マイクロプロセッサの詳細設計, 情報処理学会第43回全国大会, 2Q-12 (1991)。