

PARTHENON を用いた計算機アーキテクチャ教育

仲野 巧[†] 木村 勉[†] 前原健二^{††}
塩見彰陸^{†††} 今井正治^{†††}

[†] 豊田工業高等専門学校

^{††} 津山工業高等専門学校

^{†††} 豊橋技術科学大学

あらまし

豊橋技科大および豊田高専での高位論理合成システム PARTHENON を用いた計算機アーキテクチャ教育について報告する。

計算機の設計のためには2種類の教育用 CPU が必要で、一つはパイプライン等の高機能なアーキテクチャに拡張するための実践コース向けの基本 RISC で、もう一つは基本的な動作の学習と命令セットを拡張するための入門コース向けの基本 CISC である。

その結果、計算機アーキテクチャ教育には、ハードウェア記述言語 (HDL) で記述する高位論理合成システムとフィールドプログラマブルゲートアレイ (FPGA) が有効となる。

キーワード

計算機アーキテクチャ教育, 高位論理合成, 教育用 RISC, 教育用 CISC, HDL, FPGA

The Education of Computer Architecture Design using PARTHENON

Takumi Nakano[†] Tsutomu Kimura[†] Kenji Maehara^{††}
Akichika Shiomi^{†††} Masaharu Imai^{†††}

[†] Toyota College of Technology

^{††} Tsuyama Technical College of Engineering

^{†††} Toyohashi University of Technology

Abstract

This paper reports on the education of computer architecture design at toyohashi university of technology and toyota college of technology using high level logic synthesis system, PARTHENON.

Two kinds of educational CPU for computer design are needed; one is the basic RISC for practical courses to extend high level architecture (for example, pipeline), and the other is the basic CISC for introductory courses to study the basic behavior of computer and extend instruction set.

Therefore it is very useful for education to use high level logic synthesis system described hardware description language (HDL) and field programmable gate array (FPGA).

Key words

education of computer architecture, high level logic synthesis, educational RISC, educational CISC, HDL, FPGA

1 はじめに

集積回路技術の進歩によりハードウェアが大規模化、複雑化、さらに高機能化し、その結果、計算機の内部構造や動作原理の基礎から新しい機能の応用までを含む計算機アーキテクチャ教育が必要である。

また、これまでハードウェアに関する教育は、論理回路、計算機構成論等の講義を基に、簡単な順序論理回路や汎用 CPU チップの実験で確認していたが、ハードウェア技術の進歩とともに新しい設計手法や設計ツール等を利用した開発環境が大幅に向上したため、回路のトップダウン/ボトムアップ設計とシミュレーションによる機能検証、および FPGA 等での試作から動作確認、性能評価までが可能になった。

本稿では、豊橋技科大および豊田高専での高位論理合成システム PARTHENON[1] を用いた計算機アーキテクチャ教育について述べる。

2 高位論理合成システム

高位論理合成システム PARTHENON は、汎用ワークステーション上でハードウェア記述言語 SFL による回路設計から動作レベルのシミュレーションと論理合成、およびマクロセルライブラリを使用したマッピングと回路最適化からネットリストや回路図の出力までが可能な NTT の VLSI 設計用高位論理合成システムである。

そこで、この高位論理合成システムを用いた計算機アーキテクチャ教育を考える。

3 教育分担

技科大と高専は、教育機関として進学の面で密接な関係があるため、継続性のある教育が必要である。

そこで、技科大では、修士/博士課程で新しい構成方式の計算機アーキテクチャ研究のための基本 RISC を設計する教育を、また高専では計算機アーキテクチャの内部動作を理

解するための基本 CISC を設計する教育を高位論理合成システムを用いて実施する。

4 技科大における教育

技科大で計算機アーキテクチャ教育を行う場合、修士/博士課程での研究の基礎になり、さらに新しい機能を取り込んだ設計が可能な拡張性のある教育を必要とする。

4.1 教育目的

高位論理合成システムを用いた計算機アーキテクチャ教育の目的は、以下のようなことが考えられる。

- 計算機アーキテクチャ構成の修得
- ハードウェア記述言語の修得
- 論理合成、回路最適化の理解
- 動作速度、消費電力等の概念の理解

4.2 教育方法

計算機アーキテクチャ教育の実験は、以下のような2つの期間に実施している。

- プロジェクト実験
学年：3年次 12月から2月まで
時間：3H×2回×8W=48H
- 大実験
学年：4年次 4月から6月まで
時間：3H×2回×5W=30H

各実験は、学生のテーマ選択により約20人が一度に実験が行える環境で実施する。

4.3 教育用 RISC

設計する教育用 RISC は、計算機の基本的な概念を含んでいるようなシンプルなアーキテクチャと命令セットを採用する。

教育用 RISC の基本機能は、以下のようである。

- RISC タイプの命令セット体系

- ハーバードアーキテクチャ
- 制御用の専用 I/O レジスタ

さらに、命令や内部バスを拡張した拡張機能を設計することも可能で、ハードウェア記述言語により数日の作業で新しい機能が実現できる仕様である。

4.4 RISC アーキテクチャ

教育用 RISC は、機能拡張が可能な基本的なアーキテクチャを採用する。

- アーキテクチャ

内部アーキテクチャは、基本的な 3 バス方式で構成し、並列動作が可能である。

外部バスは、ハーバードアーキテクチャのためアドレスバスとデータバスが分離独立している。

また、データ長は基本的な 8 ビットで構成する。

図 1 に教育用 RISC のアーキテクチャを示す。

- 命令フォーマット

命令フォーマットは、命令 4 ビットとレジスタのソースとディスティネーション指定に各 4 ビットの 1 命令 12 ビット長で構成する。

- 命令セット

命令は、オペコード 4 ビットで指定する基本的な 16 種類の命令で構成する。

図 2 に教育用 RISC の命令セットを示す。

- 命令フェーズ

命令フェーズは、パイプライン等の拡張を考慮して、命令のフェッチ、デコードとエクゼキューション等を各サイクルに分割し設計する。

拡張機能は、命令長とデータ長を 16 ビットに拡張し、さらに新しい命令を追加した仕様である。

また、その他必要と思われる命令の拡張およびパイプライン等の機構は、各設計者への課題とする。

4.5 設計結果

教育用 RISC をハードウェア記述言語 SFL で記述し論理合成した結果は、以下のようである。

● パワー	11813.1	$\mu\text{W}/\text{MHz}$
● エリア	1.46563	mm^2
● ゲート数	2854.75	gates
● SFL ステップ数	516	steps
● モジュール数	11	modules
● 動作スピード	200	nsec
● ライブラリ	vsc370	VLSI Tec.

高位論理合成システムとハードウェア記述言語を使用しているため、設計者の要求するスピードやゲート数等の条件に合うように設計し、評価することも容易に可能である。

さらに、数千ゲートの回路規模であれば、FPGA の一つである Crosspoint[2] で回路を試作することによって、設計から動作確認と性能評価までの大規模な実験も可能である。

4.6 教育効果

高位論理合成システムを用いて教育を行うことによって、論理回路の設計が簡単に実現でき、さらに機能の検証はシミュレーションで迅速に行うことができる。

また、ハードウェア記述言語を用いることにより、新しい機能の追加/拡張等の変更が容易に実現できる。

従って、各種のアーキテクチャを記述して最適な選択が可能となる。

しかし、ハードウェア記述言語の記述と実際の論理合成した結果の回路が一致しないハードウェア設計者には、言語の最適化のためにハードウェア教育が必要である。

5 高専における教育

高専で計算機アーキテクチャの基礎教育を高位論理合成システムを用いて行う場合、どのような CPU を教育用として使用するかが問題となる。

そこで、設計者がそのアーキテクチャの知識を何かの目的に利用できるような標準的教育用 CISC アーキテクチャを以下のような設計目標で検討する。

- 身近で知識の再利用が可能
- シンプルな命令セット体系
- 基本的な動作の理解が可能
- 機能の拡張が可能

今回、このような点を考慮して、標準となる教育用 CISC TEST Chip を高位論理合成システムを用いて設計する。

5.1 教育用 CISC

教育的な配慮から、教育用 CISC として通産省認定の情報処理技術者試験で規程している仮想計算機 COMET を設計の対象とする。

16 ビット CPU としては、CISC タイプのため命令や内部処理は複雑だが、アセンブラの仕様と機能／動作が規程されているため、統一した設計を行うことができる。

また、命令を拡張することで新しい機能を付加することも可能である。

5.2 CISC アーキテクチャ

教育用 CISC のアセンブラ命令は、CASL で定義されているため、命令フォーマットとオペコード、レジスタ指定の割当、およびアーキテクチャ設計のみで実現が可能である。

また、CPU のデバッグは、外部端子からデバッグモードを指定し内部レジスタのリード／ライトを実行することにより、動作テストを外部 CPU から可能とするテスト容易化も実現する。

- アーキテクチャ
内部アーキテクチャは、単純な 2 バス方式で構成し、並列動作を行わない単純なレジスタトランスファの組み合わせで実現する。

図 3 に教育用 CISC のアーキテクチャを示す。

- 命令フォーマット

命令フォーマットは、16 ビットの命令を op と GR, XR にそれぞれ 8, 4, 4 ビットに分割する。

図 4 に教育用 CISC の命令フォーマットを示す。

- 命令セット

命令はオペコード 8 ビットを上位 4 ビットのグループと下位 4 ビットの機能に分割し、基本的な 24 種類の命令で構成する。

図 5 に教育用 CISC の命令セットを示す。

- 命令フェーズ

命令フェーズは命令のフェッチとエクゼキューションの可変長フェーズで処理し、各フェーズは 2 クロックで実行する。

図 6 に教育用 CISC の命令フェーズを示す。

汎用レジスタ以外の内部レジスタとしてスタックポインタ SP, 割り込みベクタレジスタ VR, フラグレジスタ FR およびプログラムカウンタ PC が汎用レジスタと同様の命令で参照できる構成とする。

外部信号として、非同期バスで動作する基本的なメモリ信号と、割り込みをエッジで検出する割り込み信号を付加する。

5.3 講義方式

講義は、次のような手順を繰り返し、担当者を決めて順番に行う方式とする。

- 講義
SFL の説明と演習課題の説明
- 演習
自由時間の演習と結果の出力
- 報告
結果発表と質疑応答

各担当者は、演習課題を一つのモジュールとして実現し、シミュレーションによりその

動作を確認し、設計したモジュールについて OHP 等で発表を行う。

また次の担当者は、必要な既存のライブラリを利用しながら新しい課題を設計するような、モジュールの階層設計で一つの機能を実現する。

グループは、端末の数から 10 班 4 人に分け、演習課題は 2 人で実行し、結果を最初の担当者が発表し、もう一人の担当者は、次の課題を別の担当者と共同で演習することにより、常に一人はオーバーラップして、前の操作方法とモジュールの内容を理解しながら演習を実施する。

さらに、10 グループが同一機能で実現方法の違う 10 種類の CPU を設計するため、最終的にパワー/エリア/ゲート/スピード等を比較する事も可能である。

5.4 演習ノート

計算機アーキテクチャ教育の講義を効率よく実現するためには、演習のためのノートが必要となる。

これは、SFL の構文と文法を項目別にまとめて適切な説明とサンプル、演習が対になっているようなテキストであり、この演習ノートが最も重要であるが、試行錯誤的に基本的な SSI/MSI を実現するモジュール設計から、ボトムアップ形式で設計済みのモジュールライブラリを利用しながら階層構造で教育用 CISC の設計ができるように作成する。

5.5 動作確認と評価

計算機アーキテクチャの設計は、ワークステーションでハードウェアを記述し、シミュレーションで機能を検証するようなソフトウェア的な環境で実現できる。

そのため、ハードウェア記述と実際の論理回路とが対応するように、現実のチップで動作を確認できるような環境が必要となる。

最近、ユーザで再書き込み可能な大規模 FPGA があるので、評価用として使用する。

回路規模は、5000 から 10000 ゲートぐらいだが、設計した回路が動作することによる興味と理解の手助けになると考える。

現在、FPGA としては、XILINX の LCA[3] を動作させており、また ALTERA の FLEX も検討中である。

最終的には、設計した CPU が直接そのまま動作することは望ましいが、設計途中の各基本機能を動作確認することでも、教育的にたいへん有効である。

6 おわりに

本稿では、PARTHENON を用いた計算機アーキテクチャ教育について現状と今後の予定について述べた。

計算機アーキテクチャの教育には、高位論理合成システムおよび FPGA 等はたいへん有効であり、記述しやすい HDL とシミュレーション環境で迅速に機能の検証と評価を行うことで、早期に教育の効果が現れ、いろいろと興味を持ちながら計算機アーキテクチャの基礎の理解からさらに新しい機能のアーキテクチャ設計も可能になる。

また、演習で設計したハードウェア回路が簡単に FPGA で動作確認できたり、教育用 RISC/CISC がレイアウトされ、現実のチップで動作すれば、さらに教育的な効果が得られると考える。

今後とも、計算機アーキテクチャ教育が実施できる環境をさらに充実させていきたいと考える。

謝辞 PARTHENON を用いた計算機アーキテクチャ教育に関係した今井研究室諸氏に感謝する。

参考文献

- [1] NTT データ通信 : PARTHENON User's Manual (1990)
- [2] Crosspoint Solutions, inc. : Crosspoint Design System (1993)
- [3] Xilinx, Inc. : Programmable Gate Array Data Book (1991)

Instruction memmory

RISC ARCHITECTURE

Instruction Type

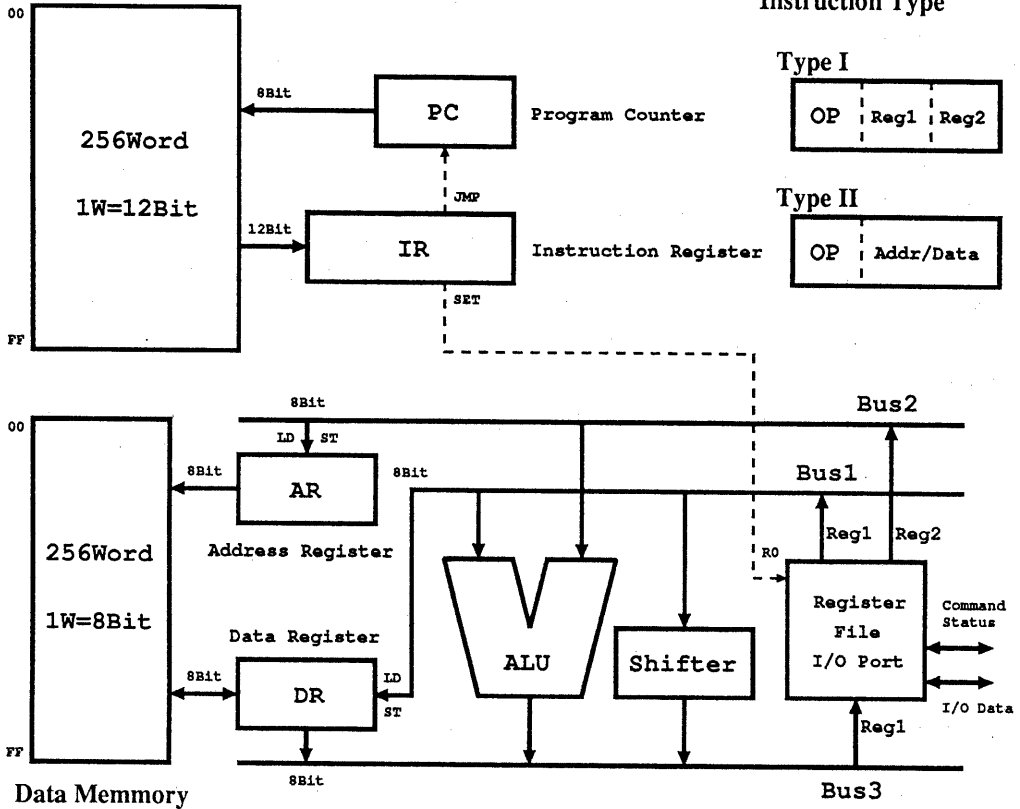
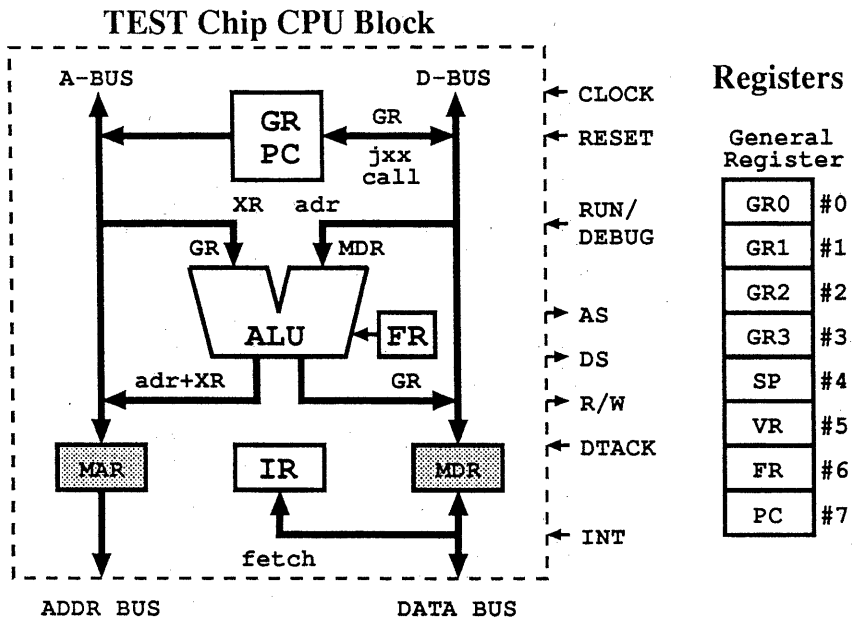


図 1: 教育用 RISC のアーキテクチャ

Instruction Set

op	Type I		op	Type II	
0	NOP	% no operation	C	JMP addr	% jump
1	LD Reg1, Reg2	% load	D	JNZ addr	% jump on not zero
2	ST Reg1, Reg2	% store	E	JNC addr	% jump on not carry
3	MV Reg1, Reg2	% move	F	SET data	% set data
4	ADD Reg1, Reg2	% addition			
5	SUB Reg1, Reg2	% subtract			
6	LSR Reg1, Reg2	% logical shift right			
7	LSL Reg1, Reg2	% logical shift left			
8	AND Reg1, Reg2	% and			
9	OR Reg1, Reg2	% or			
A	EOR Reg1, Reg2	% exclusive or			
B	NOT Reg1, Reg2	% not			

図 2: 教育用 RISC の命令セット



The Educational Standard and Testable Chip

図 3: 教育用 CISC のアーキテクチャ

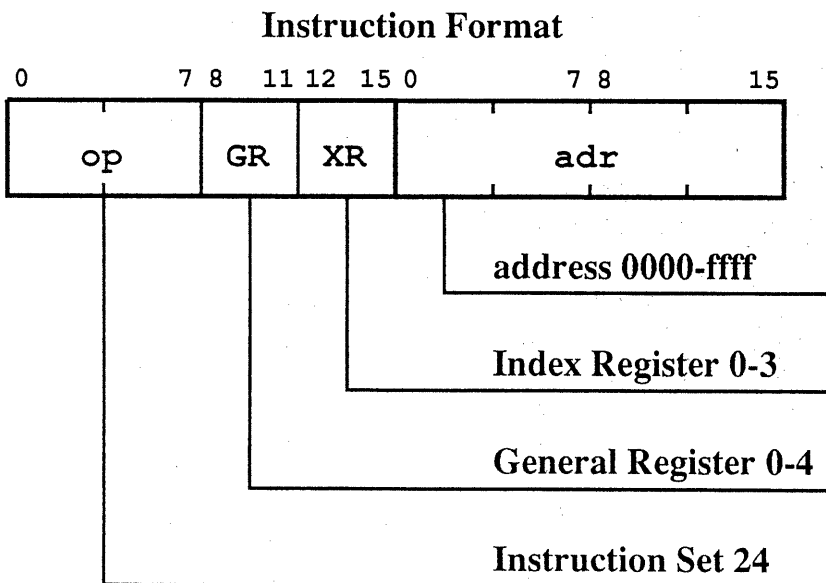


図 4: 教育用 CISC の命令フォーマット

Instruction Set

op	GR	XR	adr		
00	-	-	----	NOP	% no operation
10				LD GR,adr[,XR]	% load
11				ST GR,adr[,XR]	% store
12				LEA GR,adr[,XR]	% load effective address
20				ADD GR,adr[,XR]	% add arithmetic
21				SUB GR,adr[,XR]	% subtract arithmetic
30				AND GR,adr[,XR]	% and
31				OR GR,adr[,XR]	% or
32				EOR GR,adr[,XR]	% exclusive or
40				CPA GR,adr[,XR]	% compare arithmetic
41				CPL GR,adr[,XR]	% compare logical
50				SLA GR,adr[,XR]	% shift left arithmetic
51				SRA GR,adr[,XR]	% shift right arithmetic
52				SLL GR,adr[,XR]	% shift left logical
53				SRL GR,adr[,XR]	% shift right logical
60	-			JPZ adr[,XR]	% jump on plus or zero
61	-			JMI adr[,XR]	% jump on minus
62	-			JNZ adr[,XR]	% jump on non zero
63	-			JZE adr[,XR]	% jump on zero
64	-			JMP adr[,XR]	% unconditional jump
70	-			PUSH adr[,XR]	% push effective address
71		-	----	POP GR	% pop up
80	-			CALL adr[,XR]	% call subroutine
81	-	-	----	RET	% return from subroutine

図 5: 教育用 CISC の命令セット

Instruction Phase

fetch	p0	p1	p2
NOP/noJUMP	PC++ → MAR	* DATA → IR	* DATA → MDR
other		PC++ → MAR	

execution	p3	p4	p5
LOAD	adr+XR → MAR	* DATA → MDR	MDR → GR
STORE	adr+XR → MAR	GR → MDR *	
LOAD-EA	adr+XR → GR		
ARITHMETIC	adr+XR → MAR	* DATA → MDR	GR op MDR → GR
LOGICAL			
COMPARE	adr+XR → MAR	* DATA → MDR	GR op MDR
SHIFT	adr+XR → MDR	GR sh MDR → GR	
JUMP	adr+XR → PC		
PUSH	--SP → MAR	adr+XR → MDR *	
POP	SP++ → MAR	* DATA → MDR	MDR → GR
CALL	--SP → MAR	PC → MDR *	adr+XR → PC
RETURN	SP++ → MAR	* DATA → MDR	MDR → PC

*:data ready wait

図 6: 教育用 CISC の命令フェーズ