N-021

8 命令からなる命令セットアーキテクチャを用いたコンピュータアーキテクチャ教育教材

Educational Materials for Computer Architecture using Instruction Set Architecture Equipping Eight Instruction

寺本 圭吾 †　　弘中 哲夫†
Keigo TERAMOTO　Tetsuo HIRONAKA

1. はじめに

日々の情報技術の進歩により大学教育で教えるべき情報技術が年々増加している．その状況に対応するため，大学学部学生の情報技術者の教養として知っておくべき基本的な事柄はより短時間で本質的な事だけに絞って学習させる必要があり，コンピュータアーキテクチャ教育もその学習の一つである．高位合成技術などのハードウェア設計技術の進歩により従来ソフトウェア記述言語として扱う C 言語などがハードウェア設計に用いられる．その結果，ソフトウェア設計とハードウェア設計の間の境界が徐々になくなり，コンピュータアーキテクチャの本質的な理解が重要となる．そのため，学部教養教育としてのコンピュータアーキテクチャ教育においても自らプロセッサを設計できる学習が必要である．そこで，我々は命令セットアーキテクチャを 8 命令と大幅に絞ることで，短時間でのプロセッサの動作の理解を実現しつつ，同じ命令セットでコンパイラやオペレーティングシステムなどのシステムソフトウェアの学習を可能にする Temple を開発した．本稿では，Temple を用いたプロセッサの動作原理の理解だけでなく，アセンブリ教育やプロセッサの改良まで対応した教育教材について述べる．

2. 教育カリキュラム

図 1 に Temple で実現するコンピュータアーキテクチャ教育カリキュラムを示す．以下の節において各項目の詳細について紹介する．

2.1 アセンブリプログラミング

命令セットアーキテクチャの理解のためにアセンブリプログラミングを行う．高級言語で一般的に使用される構文をアセンブリプログラミングで表現する．具体例として for 文，if 文，配列操作，関数呼び出し，スタック操作などの記述方法をアセンブリプログラミングで学ぶ．

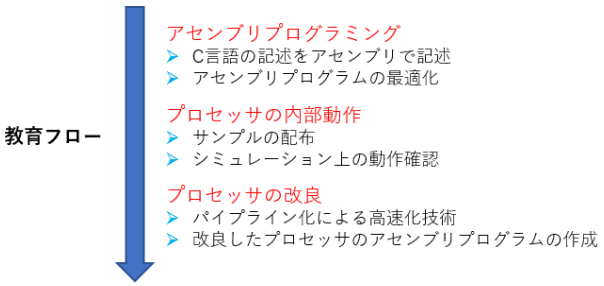

図 1　コンピュータアーキテクチャ教育の教育カリキュラム

2.2 プロセッサの内部動作理解

プロセッサの内部構造と各命令の内部動作を理解する．サンプルのプロセッサ，回路図，アーキテクチャの概要を提示して説明を行う．各命令毎に回路図のデータパスを確認しながら学習する．次にデータパスを理解した上でデコーダやステートマシンの制御回路をハードウェア記述言語で作成する．

2.3 プロセッサの改良

サンプルから改良して，パイプライン化に対応したプロセッサを作成する．パイプライン化に必要なステージ分割を行い，資源や命令のハザードをハードウェアとソフトウェアの両面から対処できる方法を理解する．実装したパイプラインプロセッサの各命令の動作テストを行う．アセンブリプログラミングを行い性能を評価する．

2.4 学習における命令セットの影響

前節で示す教育内容は特段新しいものではなく，従来から行っているものである．しかし，従来コンピュータアーキテクチャ教育で使用される MIPS や ARM の命令セットアーキテクチャ [1] では，命令セットのサブセットを用いてもアセンブリプログラミングに対応する命令セットであればある程度命令数が必要となる．また，内部動作パターンは命令数に比例した数を実装する必要があり，命令数が増加すると多大な時間を要する．一方，mcpu[2] に見られるように命令セットを固定長にし，かつ，4 命令に制限すると内部動作理解は容易になる．しかし，mcpu の命令セットでコンパイラなどのシステム

† 広島市立大学大学院 情報科学研究科

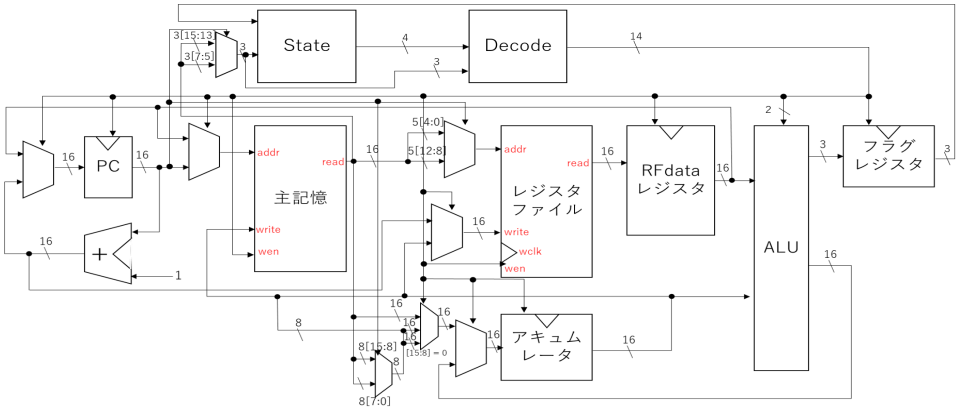

図 2　Temple のブロック図

表 1　Tempe の命令セット

オペコード	命令操作
NOR	$Acc = $Acc nor $mX
ADD	$Acc = $Acc + $mX
LD	$Acc = Mem[$mX]
MOVE	$mX = $Acc
SD	Mem[$mX] = $Acc
SETI	$Acc = Immediate
JL	$mX = PC if (cond) PC = $mY
SRL	$Acc = $Acc >> 1

ソフトウェアの学習に用いるのは困難であり，構造の簡略化を進めると一般に大きなプログラムを格納するメモリ空間を設けづらい (mcpu のメモリは 64Byte)．本稿で提案する命令セットアーキテクチャ Temple は，前節で述べた教育カリキュラムを 15 コマの講義時間で行うことを目標に命令数と各命令の機能を決めたものである．

3. Temple アーキテクチャ

Temple は少ないゲート数で実用的なプログラミングをすることを目的に開発された命令セットアーキテクチャ Pilaf[3] を，コンピュータアーキテクチャ教育向けに改良した 8 命令を備える 64KByte のメモリ空間を備える 16 ビットプロセッサである．図 2 に教育教材のサンプルとして提供する Temple のブロック図を示す．以下，Temple アーキテクチャの概要を示す．

3.1　レジスタ

Temple アーキテクチャはアキュムレータ，フラグレジスタ，32 本の汎用レジスタの 3 種類のレジスタを備える．フラグレジスタは ALU を用いた演算の結果によりゼロ，負，キャリーを示し，条件分岐の状態として使用される．汎用レジスタを 32 本 (内 3 本が 0，1，-1 の定数を格納する定数レジスタ) 備え，演算，分岐命令，ロード/ストアの引数として使用されるほか，必要に応じて複雑な機能を実現するための疑似命令 [4] の作業用レジスタとして使用される．

3.2　命令セット

表 1 に Temple の命令セットを示す．$Acc はアキュムレータを表している．$mX と$mY はレジスタを表しており，X と Y には 0〜31 番の値が入る．Temple の命令セットは 8 命令で構成されている．LD 命令や SD 命令による間接参照を用いることで，配列演算を実現が可能である．JL 命令を用いることで，プログラムカウンタがレジスタの値に書きかえらるため，高級言語の実装に必要なサブルーチンコールの実現が可能である．

4. おわりに

本稿では，命令セットアーキテクチャ Temple を用いたコンピュータアーキテクチャ教育の教育教材を作成した．Temple を用いた教育教材では，内部動作の理解だけでなく，アセンブリ教育やプロセッサの改良まで幅広い教育が可能となる．現在割り込みに対応した Temple プロセッサも開発中であり，今後はオペレーティングシステム教育に対応した教育教材の作成を計画している．

参考文献

[1] David Harris, Sarah Harris,“Digital Design and Computer Architecture 2nd Edition,” Morgan Kaufmann, (2012).

[2] TimBoscko,“MCPU　A Minimal 8bit CPU in a32 macrocell CPLD,” https://github.com/cpldcpu/MCPU, (2001).

[3] 米田浩貴，胡濱良樹，児島彰，弘中哲夫，“Pilaf:超小型プロセッサ IP の開発,” 電気学会 電気学会研究会資料 電子回路研究会 ECT-15-071, pp. 7-12, (2015).

[4] 寺本圭吾，胡濱良樹，窪田昌史，谷川一哉，弘中哲夫，“小規模 IoT 機器向けマイクロプロセッサ Pilaf のためのマクロアセンブラの開発,” 信学技報, vol. 116, no. 240, CPSY2016-44, pp. 11-12, (2016).