

命令セットが8命令からなる小規模CPUを対象とした C言語風コンパイラ実装

大本 裕真[†] 川端 英之[‡] 弘中 哲夫[‡]
広島市立大学 情報科学部[†] 広島市立大学 大学院 情報科学研究科[‡]

1 はじめに

情報工学を学ぶ上では、コンピュータの動作原理とシステムソフトウェアの仕組みを理解する必要がある。しかし、一般にコンピュータの動作理解教育に用いるCPUは、講義時間の制約から命令セットを簡潔にすることが求められる。一方でシステムソフトウェア教育には、命令数が多く、充実した命令セットを必要とする。これに対し我々は、厳選した8命令で両教育を実現するCPUである Temple を開発している。

本稿では、Temple 上で動作するシステムプログラムが設計可能であり、システムソフトウェア教育に適用できることを評価するために、実際に Temple 用ソフトウェア開発環境の実装を試みた結果について報告する。実装したシステムソフトウェアは、情報科学部1年生が講義の合間に、3ヶ月かけてオンデマンドでCPUアーキテクチャから学び、C言語の主要な文法をサポートするコンパイラ、アセンブラ、および、CPUシミュレータを開発したものである。

2 Temple 命令セットアーキテクチャ

Temple プロセッサは、メモリアクセスや関数呼び出しを可能にする8個の命令からなる表1の命令セットを備えている。このうち、JL命令ではcond部分で指定する条件でZNCフラグを判定し、分岐の可否を判断する。

表 1: Temple の命令セット

オペコード	記述例	動作	変化するフラグ
NOR	NOR \$mX	\$Acc = \$Acc NOR \$mX	Z N -
ADD	ADD \$mX	\$Acc = \$Acc + \$mX	Z N C
LD	LD \$mX	\$Acc = Mem[\$mX]	---
MOVE	MOVE \$mX	\$mX = \$Acc	---
SD	SD \$mX	Mem[\$mX] = \$Acc	---
SETI	SETI immediate	\$Acc = immediate	---
JL	JL \$mX cond \$mY	\$mX = PC if(cond)PC = \$mY	---
SRL	SRL	\$Acc = \$Acc >> 1	Z --

3 ソフトウェア開発環境の概要

作成したソフトウェア開発環境は、アセンブラ、C言語風コンパイラ、および、CPUシミュレータである。アセンブラとCPUシミュレータは、一つのプログラムとしてC++言語で実装され、コマンドライン引数により実行する処理を変更することができる。C言語風コンパイラはC言語で実装されている。また、表2は各ソフトウェア開発環境の主な処理と、その実装に必要とした行数である。次節に、各プログラムの仕様概要を示す。

表 2: 各ソフトウェア開発環境の処理と行数

(a) アセンブラ・CPUシミュレータ		(b) コンパイラ	
コマンドライン引数解析	55行	字句解析	261行
字句解析	70行	構文解析	615行
シミュレータ	217行	コード生成	440行
アセンブラ・逆アセンブラ	497行	総行数	1316行
総行数	839行		

3.1 アセンブラ・CPUシミュレータ

アセンブラ・CPUシミュレータには、アセンブリ言語を機械語に変換するアセンブラモードと、機械語をアセンブリ言語に変換する逆アセンブラモードがある。また、アセンブラモード

An implementation of a C-like language compiler for a small CPU with 8 machine instructions

[†] Yuma Omoto, Hiroshima City University

[‡] Hideyuki Kawabata, Tetsuo Hironaka, Hiroshima City University

では 11 個の疑似命令 (sub, push, pop, nop, nopi, jmp, load, save, halt, call, ret) と Temple の 8 つの命令と合わせて、19 個の命令が使えるアセンブラとなっている。

アセンブラ・CPU シミュレータは、大きく 4 つの処理から構成されている。CPU シミュレータは表 2 (a) に示すように、シミュレータ実装に必要な記述部分はわずか 217 行に抑えられ、コンパクトにコーディングすることができた。これは Temple が非常に小規模な命令セットを持つ CPU であるためである。なお、アセンブラ・CPU シミュレータの製作期間は約 1 カ月である。

3.2 C 言語風コンパイラ

C 言語風コンパイラ (Temple_cc) は、現時点で if, for, while, return 文や、再帰呼び出し可能な関数、ポインタ演算、変数は int(16 ビット符号付き) と int *(16 ビット) に対応している。

実装は x86-64 命令セットを対象とした文献「低レイヤを知りたい人のための C コンパイラ作成入門」[1] を参考にした。そのため、Temple の命令セットでは必要な命令が不足した。そこで、Temple_cc 実装のために、使用されている x86-64 命令に対応する疑似命令を Temple の命令セットで作成し、それを用いてコンパイラのコード生成系を作成した。実装に用いたコード行数を表 2 (b) に示す。なお、Temple_cc の製作期間は約 2 カ月である。

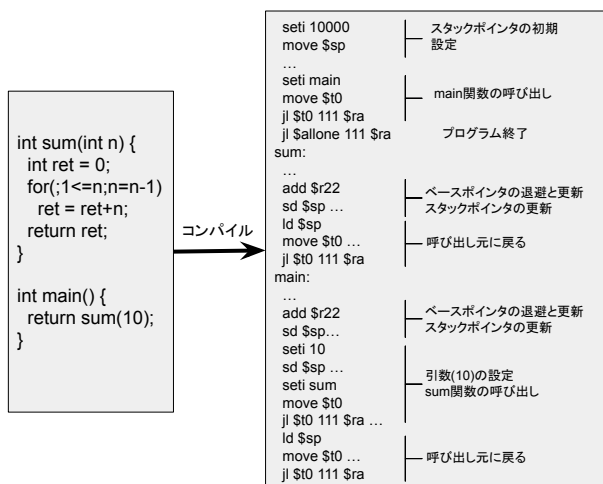


図 1: C プログラムとコンパイラ出力

4 動作例

実際の C プログラムと、コンパイラ出力を図 1 に示す。コンパイルには関数呼び出しと for 文を使用して 1~10 までの総和を求めるプログラムを用いた。また、再帰処理を行うプログラムでも正常に動作することを確認している。

5 まとめ

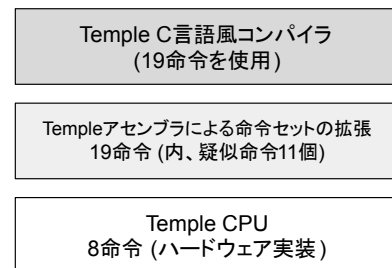


図 2: 疑似命令を用いた命令セットの拡張

Temple CPU の厳選した命令セットを用いることで、図 2 のようにコンピュータの動作理解を容易に行うことができ、更に疑似命令を活用した 19 命令によるアセンブラプログラミングや C 言語風コンパイラを利用したシステムソフトウェアの設計開発を行うことができる事が実証できた。今後の課題としては、コンパイラを拡張し、配列、グローバル変数などを取り扱えるようにすること。また、コンパイラにコード最適化機能を実装することである。なお、本稿で制作しているコンパイラは以下の URL に公開されている。

https://github.com/Uma821/Temple_cc

参考文献

- [1] Rui Ueyama, “低レイヤを知りたい人のための C コンパイラ作成入門”, <https://www.sigbus.info/compilerbook>, 2020 年 3 月 16 日.
- [2] 寺本 圭吾, 弘中 哲夫, “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,” 第 17 回情報科学技術フォーラム FIT2018, 第 4 分冊, pp.311-312, 2018 年 9 月 19 日.