組込み開発向けスクリプト言語mruby/cを支援するハードウェアアクセラレーションの開発

前田 洋征[*1]　田中 和明[*2]

Hardware Acceleration for Embedded Scripting Programming Language ‘mruby/c’

Hiroyuki Maeda, Kazuaki Tanaka

スクリプト言語はプログラムを短く記述でき，読みやすい特徴から開発効率に優れているが，言語処理系がプログラムを実行する仕組みになっているため，処理速度が遅い問題がある。そこで本研究では，小型な組込み開発向けスクリプト言語mruby/cを対象にボトルネック処理の解析を行い，ボトルネック処理を支援するハードウェアアクセラレーションを設計した。設計したハードウェアアクセラレーションと連携して動作するmruby/cの実行環境になるSoC（System On a Chip）をFPGAに構築し，性能評価した結果，約1.4倍の高速化を実現した。

1 はじめに

スクリプト言語は，自然言語に近い表現でプログラムを記述でき，可読性も高いため開発効率に優れるプログラミング言語である。近年ではWebアプリケーション開発だけでなく組込み開発でも利用されている。しかしながら，スクリプト言語は仮想マシン等の言語処理系がプログラムを実行するインタープリタ方式であるため，処理速度が遅い問題がある。そこで本研究では，IoT 等の組込み開発で利用されている mruby/c を対象に仮想マシンのボトルネック処理を解析し，解析結果からボトルネック処理を支援するハードウェアアクセラレーションを設計した。設計したハードウェアアクセラレーションと mruby/c の実行環境を FPGA に構築し，仮想マシンが各ハードウェアと連携してプログラムを実行する SoC を構築し，性能評価を行った。

2 mruby/c

mruby/cは，Rubyの開発効率の良さを引き継ぎつつ，組込み開発向けに省メモリ化したスクリプト言語である。特に小さなデバイスを用いたIoT開発を対象にしており，ワンチップマイコンなどの50 KB以下のメモリ容量が厳しい環境でも動作する。また，独自機能として複数のプログラムを同時に実行するマルチタスク機能が備わっている[1)]。実行時は，プログラムをバイトコードに変換し，仮想マシンがバイトコードを読み取ることで実行する仕組みになっている。

*1 機械電子研究所
*2 九州工業大学情報工学部

3 mruby/c仮想マシンのボトルネック処理の解析

本研究では，mruby/cの仮想マシンにおける遅延の要因を解析し，それらを支援するハードウェアアクセラレーションを設計した。解析には，mruby/cの実行環境を構築したFPGAを使用し，仮想マシンに負荷がかかる「10までのフィボナッチ数列を計算するプログラム」を実行して各処理の頻度及び合計時間を測定するプロファイルを実施した。結果を表1に示す。

表1 プロファイル結果

処理内容	関数	処理回数 [回]	合計処理時間 [ミリ秒]
メモリ確保及び解放	mrbc_init_alloc	1	2.97
	mrbc_raw_alloc	283	823.53
	mrbc_raw_alloc_no_free	21	62.58
	mrbc_raw_free	416	267.81
バイトコードの識別・取得	mrbc_load_mrb	1	3.92
命令実行	mrbc_vm_run	1206	771.84
クラス・メソッド定義	mrbc_init_class	1	515.72
	mrbc_define_class	27	101.52
	mrbc_define_method	164	485.44
メソッド呼び出し	mrbc_get_irep_symbol	136	3.81
	str_to_symid	321	21.19
	find_method_by_class	110	7.15
タスク管理	mrbc_create_task	1	4.54
	mrbc_init_tcb	1	0.02
参照カウント	mrbc_dup	329	1.31
	mrbc_release	1958	9.79

表1の結果から処理頻度が高く，合計の処理時間が長い処理は，「メモリ確保及び解放」「命令実行」「クラス・メソッド定義」「メソッド呼び出し」である。これらの処理が仮想マシンにおけるボトルネックになっていると判明した。そこで，「メモリ確保及び解放」を支援するためのMalloc回路，「命令実行」を支援するLoad_Decode回路，「クラス・メソッドに関する処理」を支援するSymbol回路の設計を行った。

4 開発したハードウェアアクセラレーションの構成

開発環境には，Intel製FPGAを搭載したDE0-NanoボードとNios IIシステムを使用した。Nios IIシステムは，FPGAを書き換えてソフトコアプロセッサを構築し，メモリや周辺回路を組み合わせて，独自にSoCを構築できるシステムである。図1にシステム構成を示す。

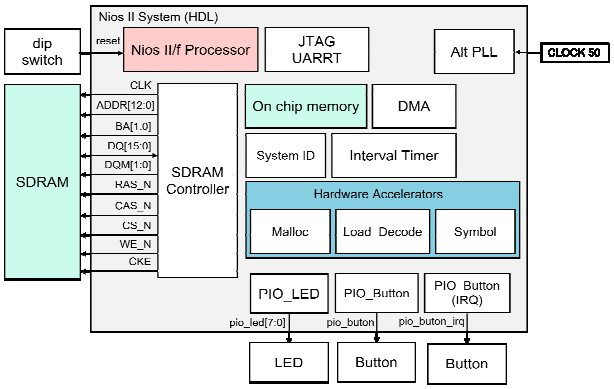

図 1 FPGA に構築したシステム構成

本システムは， Nios II/fプロセッサとメモリであるSDRAMとOn chip memory，その他の周辺回路と独自設計したMalloc回路，Load_Decode回路，Symbol回路で構成される。mruby/cの仮想マシンが各回路と連携してプログラムを実行する流れを図2に示す。

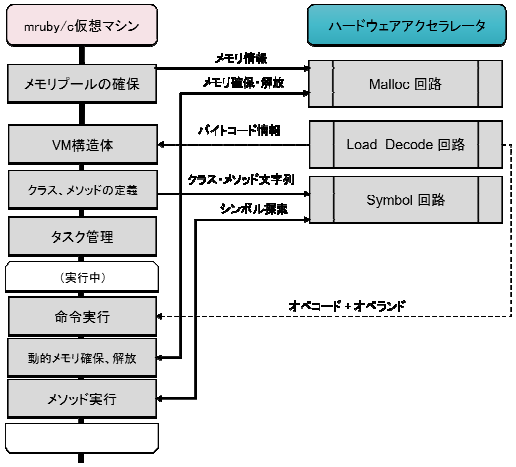

図 2 仮想マシンと各回路の動作

Malloc回路がサイズ毎の空きメモリ情報を管理しており，仮想マシンの処理中にメモリが必要な時は，Malloc回路が要求サイズから割り当てるメモリブロックを算出する。また，Load_Decode回路がバイトコードの取得と命令フェッチ・デコードを行い，仮想マシンは結果を受け取って命令を実行する。Symbol回路は，クラス・メソッドに関する処理の支援を行う。シンボルとは，クラス・メソッド等の文字列のことである。クラス・メソッドの定義では，Symbol回路がシンボルからハッシュ値の計算とシンボルIDの算出，シンボルテーブルの登録を行う。メソッド呼び出しでは，Symbol回路がシンボルからハッシュ値の計算を行い，シンボルテーブルのハッシュ値が合致するシンボルIDを探索し，仮想マシンがこのシンボルIDに対応する処理を行うことでメソッドが実行される。

5 性能評価

10までのフィボナッチ数列を計算するプログラムと50個のランダムな数例をマージソートするプログラムを実行し，処理速度を測定した。結果を表2に示す。

表2 処理速度の測定結果

ベンチマークプログラム	仮想マシンのみ[ミリ秒]	HW有り[ミリ秒]	向上率[倍]
フィボナッチ数列計算(10)	1175.41	836.39	1.41
マージソート(50)	4489.52	3128.65	1.43

mruby/c仮想マシンにハードウェアアクセラレーションを適用することで異なる2種類のプログラムにおいて，約1.4倍の高速化を実現した。

6 まとめ

本研究では，FPGAボードに組込み開発向けスクリプト言語mruby/cの仮想マシンを支援するハードウェアを設計・実装し，mruby/cが専用ハードウェアと連携してRubyプログラムを実行するSoCを構築した。評価の結果，プログラムの種類に依存せず，必ず実行されるボトルネック処理を対象にハードウェア化したことで，処理速度の高速化を実現した。

7 参考文献

1) Kazuaki Tanaka,Hiroyuki Maeda,Hirohito Higashi, Concurrent execution in Scripting Programming Language 'mruby', Computational Science and Its Applications-ICCSA2018, pp.136-146

8 掲載論文誌

日本ソフトウェア科学会 コンピュータソフトウェア，2024年41巻2号，pp. 60-79