1A-02

スクリプト言語を支援するハードウェアアクセラレータの実装

前田洋征† 田中和明‡
九州工業大学 情報工学府† ‡

1 はじめに

IoT 技術を活用した組込みシステムが多く登場している．IoT では実社会の課題を解決することが重視され，より柔軟な組込みシステムの開発が必要となり開発効率の向上が求められる．スクリプト言語は，C 言語よりも容易で柔軟なプログラミングが可能で，IoT 開発に利用することで開発効率の向上が期待できる．しかしながら，仮想マシンによって動作するスクリプト言語は C 言語と比べ実行速度が遅いという問題がある．この問題を解決するために，IoT などの小型端末向けに開発されたスクリプト言語 mruby/c を対象に，仮想マシンを支援するハードウェアアクセラレータを実装した[1]．仮想マシンの各処理ごとにアクセラレータを実装し，アクセラレータ間は協調して動作するようにハードウェアの構成を最適化した．

2 スクリプト言語

スクリプト言語は，プログラムを短く記述でき可読性が高いため，開発効率やメンテナンス性に優れるプログラミング言語である．Ruby や Python，JavaScript などが挙げられ，特にハードウェア資源が豊富な Web アプリケーション開発で広く利用されているが，近年では IoT アプリケーションの開発に利用する場面も増えてきている．スクリプト言語は，ハードウェア制御や複雑なアルゴリズムをより短く記述できるので，プログラムを作成してからハードウェアを動かし，動作結果を元にプログラムを改良して再度試すことを繰り返す IoT 開発の作業を効率的に進めることができる．

2.1 mruby/c

mruby/c は，小型なデバイスを用いた IoT 開発を想定して作られたスクリプト言語である．Ruby の特徴を引き継ぎつつ軽量化されており実行に必要なメモリは，50 KB 未満なため，ワンチップマイコンなどメモリ制約が厳しい環境でも動作可能になっている． mruby/c は IoT 開発で実用化されており，醸造業向け温湿度管理や水田の水位計測システムなど，実社会の課題解決に mruby/c が実際に使用されている[2]．

mruby/c の実行は，予めユーザプログラムをコンパイルし専用の機械語であるバイトコードに変換し，マイコンなどの実行環境で仮想マシンがバイトコードを逐次読み取り実行することでプログラムが動作する仕組みになっている．面倒なメモリの管理は仮想マシンが行い，型宣言やハードウェアに依存する処理などは仮想マシンによって補完されるため，ユーザは容易なプログラミングが可能になる．しかしながら，スクリプト言語は，C 言語などのコンパイラ型言語と比べると実行速度が遅くなる問題がある．

3 ハードウェアアクセラレータ

本研究では，スクリプト言語を高速化するために仮想マシンを支援するハードウェアアクセラレータを実装した．ハードウェアアクセラレータの実装には，Intel FPGA を搭載した Terasic 社製の開発ボードの DE0-Nano と Nios II システムを使用した．Nios II システムは，FPGA の論理回路やメモリブロックからソフトコア・プロセッサを構成し，メモリや IP(Intellectual Property)コア，任意のペリフェラルを組み合わせることで独自のシステムを構築できる[3]．

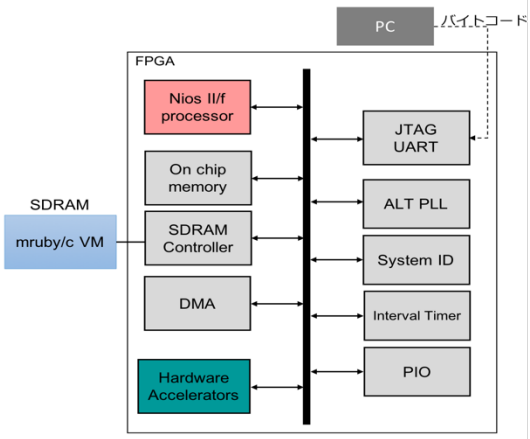

図1 NiosII システムの構成

Implementation of Hardware Accelerators for Scripting Programming Languages
†Maeda Hiroyuki, Kyushu Institute of Technology
‡Tanaka Kazuaki, Kyushu Institute of Technology

DE0-Nano に構成した NiosII システムを図 1 に示す．本システムは，ソフトコア・プロセッサ Nios II/f と FPGA の内部メモリである On Chip Memory，外部メモリである SDRAM，プロセッサを介さずにメモリ間やメモリから周辺回路に直接データを転送できる DMA(Direct Memory Access)，その他の IP コアに加えて，仮想マシンの各処理を支援するハードウェアアクセラレータから構成されている．mruby/c 仮想マシンとバイトコードは SDRAM に書き込まれる．バイトコードは，DMA を用いてアクセラレータに転送される．

本研究では，仮想マシンの処理の中でメモリアロケーション，バイトコードの読み取り，命令デコード，シンボルテーブルの管理をそれぞれ支援するハードウェアアクセラレータを，SystemVerilog を用いて実装した．各ハードウェアアクセラレータの動作を図 2 に示す．

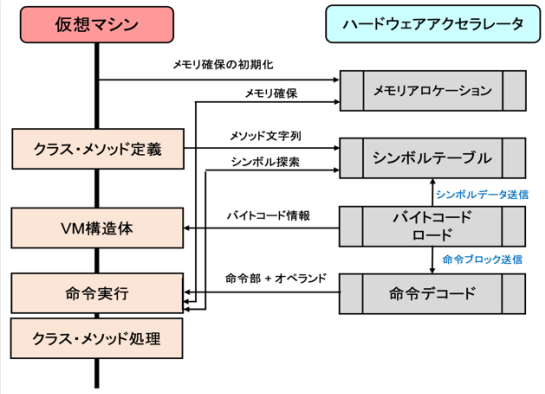

図2 仮想マシンとハードウェアアクセラレータ

仮想マシンは，実行時に動的なメモリ確保を行なっている．割り振るメモリは，ハードウェアアクセラレータが要求サイズとサイズごとの空き領域の情報を元に計算し，割り振るメモリ領域を決定する．ハードウェアアクセラレータがバイトコードの読み取りを行い，バイトコードに含まれる全体のサイズや命令数，命令実行に必要な情報，命令ブロック，メソッド文字列(シンボル)などを仮想マシンの各情報を保持する VM 構造体のメンバー変数へ送信する．バイトコードを読み取るアクセラレータは，バイトコードの命令ブロックとシンボルデータを，それぞれを処理するアクセラレータに送信する．命令の実行時には，アクセラレータが命令ブロックの命令部とオペランドを読み取り仮想マシンに送信する．仮想マシンは，アクセラレータから受信した命令部とオペランドを元に対応する処理を行う．

仮想マシンには，Ruby メソッドに対応する関数が実装されており，メソッド文字列を元にシンボルテーブルが作成される．シンボルテーブルは木構造になっており，メソッドを実行する際は，このシンボルテーブルからメソッドに対応する関数を探索し実行することで，メソッドの実行が行われる．本研究では，ハードウェアアクセラレータがシンボルテーブルの作成とメソッド実行の際のシンボル探索を行うことで，より高速にメソッドの実行が可能になった．

以上のように仮想マシンとハードウェアアクセラレータが連携して動作することで，より高速なプログラムの実行を実現した．

4 性能評価

ハードウェアアクセラレータの効果を検証するために，仮想マシンのみで実行した場合とアクセラレータを使用して実行した場合の実行時間を計測した．ベンチマークには，10 までのフィボナッチ数列を計算するプログラムを使用した．

表 1 フィボナッチ数列計算の実行時間

仮想マシンのみ	アクセラレータを使用
4,182 [ms]	3,718 [ms]

計測の結果からハードウェアアクセラレータを使用することで実行速度の高速化を確認した．

5 まとめ

本稿は，IoT などの小型端末向けスクリプト言語である mruby/c の仮想マシンを支援するハードウェアアクセラレータの実装を述べ，性能評価の結果から実行速度の高速化を実現したことを示した．

参考文献

[1] Kazuaki Tanaka, Hiroyuki Maeda, Hirohito Higashi ： Concurrent execution in Scripting Programming Language ‘mruby’, Proc. 18th International Conference on Computational Science and Its Applications (ICCSA 2018), 2018, pp.136--146.

[2] 神崎 映光, 北野 数馬, 吉崎 翼：組込み機器向け言語 mruby/c により動作する小規模農業支援のための無線センサネットワークの構築について，情報処理学会第 22 回コンシューマ・デバイス＆システム (CDS) 研究発表会，2018, Vol.2018-CDS-22 No7.

[3] Hiroyuki Maeda, Kazuaki Tanaka: Hardware Acceleration of Language Processing in Scripting Programming Languages, Proc. 19th International Conference on Computational Science and Its Applications (ICCSA 2019), 2019, pp. 407-416.