2020 年度 卒業研究発表 梗概

電気学科

学籍番号 氏　名	V17126　松川 達哉	指導教員	藤枝 直輝
題　目	ソフトプロセッサでシステムコールの検証を容易に行う方法の提案		

1　はじめに

FPGA（Field Programable Gate Array）に実装して使用するプロセッサであるソフトプロセッサには，多くの利点がある．ソフトプロセッサの検証では，シミュレータとプロセッサの実行結果を比較することが行われる．ファイルなどを扱う実用的なアプリケーションの検証では，シミュレータではシステムコールを模倣することがしばしば行われるが，この方法をソフトプロセッサに適用するのは困難であった．本研究では，Verilator を用いて，ソフトプロセッサでシステムコールの検証を容易に行う方法を提案する．

2　背景

プロセッサのアーキテクチャは，プロセッサが備える命令を規定する命令セットアーキテクチャ（ISA）と内部動作を規定するマイクロアーキテクチャに分けられる．プロセッサシミュレータはプロセッサの動作を模倣するもので，プログラミング言語によって書かれる．ソフトプロセッサは通常ハードウェア記述言語によって書かれる．ソフトプロセッサの検証では，シミュレータとプロセッサで実行結果を比較することが行われ，例えば，MIPS システムシミュレータ SimMips [1] とそれをもとにしたプロセッサとの間でも相互検証が行われている．

OS の機能の 1 つはインターフェイスの提供であり，そのうち API は，OS の資源を利用するアプリケーションのために提供され，アプリケーションにとって OS への窓口の役目を果たす．UNIX の API では，個々の OS の機能はシステムコールと呼ばれ，システムコールは OS の核であるカーネルに実装されている．

3　提案手法

ハードウェア記述言語の一種である SystemVerilog を C/C++ に変換する Verilator というツールを，ソフトプロセッサへ使用する．Verilator は SystemVerilog のモジュールをクラス化でき，設計者は C++ からモジュールのクロック入力を制御することで，テストベンチを書くことができる．

本研究では，C++ テストベンチからシステムコール関数を呼ぶことで，プロセッサでシステムコールが検証できることに注目する．このとき，シミュレータも C/C++ で書かれているのであれば，シミュレータでシステムコールを模倣する記述を，できる限りプロセッサの記述にも流用したい．そのため，本研究ではこうした記述を syscall() という関数にひとまとめにする．

4　ケーススタディと評価

本研究では，公開されているプロセッサに合わせてシミュレータを作製する．オープンソースのシミュレータは高機能すぎること，本研究では RV32I が実装されていれば十分ということが理由である．RV32I が実装されていて，SystemVerilog で記述されているという理由で kronos [2] というプロセッサを選択した．

本研究では，まず，kronos のシミュレータを作製し，アーキテクチャ状態の比較によってシミュレータの動作検証を行った．kronos シミュレータは，kronos プロセッサを命令レベルでシミュレートしている．動作検証によって，kronos シミュレータは，シミュレートするプログラムに問題がないとき，kronos プロセッサと命令単位で等価な動作をすることが確認できた．

その後，今回提案した方法でシステムコールが検証できるのかどうかを確かめるために，システムコールを伴うアプリケーションをプロセッサとシミュレータで実行し，取得したアーキテクチャ状態のログを比較した．アプリケーションには，組込み向けのベンチマークソフトである MiBench の automotive カテゴリーのプログラムを使用した．6 つあるプログラムのうち，4 つで正常にプログラムが動作し，その動作がプロセッサとシミュレータとで一致していることが確認できた．また，このときの処理速度は，プロセッサでは毎秒 149〜154k 命令，シミュレータでは毎秒 215〜217k 命令であった．

参考文献

[1] 藤枝直輝，渡邉伸平，吉瀬謙二 : 研究・教育に有用な MIPS システムシミュレータ SimMips，情報処理学会論文誌，vol. 50，no. 11，pp. 2665-2676，Nov. 2009

[2] kronos，https://sonalpinto.github.io/kronos/#/，検索日 : 2021/01/25