

学籍番号 氏名	V19089 杉山 皓星	指導教員	藤枝 直輝
題目	ソフトプロセッサ相互検証における 汎用メモリアンターフェースの提案		

## 1 はじめに

ハードウェア記述言語で論理回路を記述し、FPGA などに実装するプロセッサをソフトプロセッサと呼ぶ。ソフトプロセッサは、専用回路で高速化したい処理以外の汎用的な処理を実行するためや、新しいアーキテクチャ技術を検証するために用いられる。ソフトプロセッサの開発手法には相互検証と呼ばれるものがある。これは動作を命令ごとに再現するシミュレータとテストベンチを用意し、これらの命令レベルの実行ログを比較することで行う検証方法である。高水準言語を使った相互検証で問題となるのが、システムコールの扱いである。これに対して松川ら [1] は、Verilator というツールでプロセッサのハードウェア記述を C/C++ に変換する方法を提案した。これにより、テストベンチとシミュレータとでほぼ共通のコードでシステムコールの模倣が記述できる。また、本橋の研究 [2] ではより多くのシステムコールへの対応やコードの一部の統一化に取り組んだ。

本研究では、相互検証に適したメモリモジュールとインターフェースを新たに作成することで、対応プログラムを増やし、効率的に開発できる枠組みの提供を目指す。

## 2 先行研究の問題点

松川らの研究 [1] や本橋の研究 [2] では、開発対象のプロセッサとして kronos を採用している。また、実行させるプログラムには、組込みシステム向けベンチマークの MiBench を利用している。これまでに MiBench の 6 種類のカテゴリの 1 つである Automotive カテゴリのプログラムの実行に成功している。

しかし現状の検証環境には 2 つの問題点がある。1 つ目はコマンドライン引数を要求するプログラムが動作しないことである。これは、プログラム実行中のシステムコールのみを模倣する検証環境では、プログラム実行前のロード処理を完全には実装していないためである。2 つ目は kronos テストベンチに同梱されているメモリモジュールを使用していることである。このメモリとインターフェースは kronos 向けに実装されているため、kronos 以外のプロセッサで検証する場合は仕様の違いが問題となる。

## 3 提案手法

本研究では上記の問題を解決するために、より汎用的な拡張メモリモジュールとインターフェースの実装、プログラム実行準備処理の改良を行う。

既存のメモリは SystemVerilog で記述され、トップモジュールにてプロセッサと接続している。そのため C++ 記述のテストベンチではクロック信号を駆動する処理のみを記述していた。それに対し提案手法では、拡張メモリはテストベンチ内に配列として記述し、Verilator の適用範囲はプロセッサ本体のみを含むラッパーモジュールとする。また、拡張メモリを動作させる記述をクロックの立上り前後に記述する。

また、先行研究ではメモリのテキスト領域へ実行命令列を書き込む際に先頭のアドレスを手入力していた。これをコンパイラが生成する objdump ファイルから自動設定されるように改良した。さらに、スタック領域へコマンドライン引数に相当するデータ構造を書き込み、レジスタのスタックポインタを操作することで、引数が渡された状態を再現する。

## 4 検証と評価

検証には先行研究で動作確認された Automotive カテゴリのプログラムを使用する。各プログラムで拡張メモリでも正しい動作結果が得られるか確認し、既存のメモリを使用した場合との実行速度を比較する。

評価の結果として、basicmath と bitcount では正常な動作結果が得られた。しかし、それ以外のプログラムではファイルの読み込みが正常に行われず、途中で動作が停止した。これは、システムコール read の模倣処理で、読み込んだファイルを正しくメモリに書き込めていないことが原因であると考えられる。また、実行速度は全てのプログラムにおいて既存のメモリに比べて 5~14 %程度低下した。

インターフェースの改善やコマンドライン引数を必要とするプログラムの検証などが今後の課題である。

## 参考文献

- [1] 松川達哉, 藤枝直輝, ソフトプロセッサの相互検証に関するケーススタディ, 情報処理学会研究報告 2021-ARC-244, No. 37, 2021.
- [2] 本橋一馬: Verilator を用いたソフトプロセッサの検証手法の改善と拡張, 卒業論文, 愛知工業大学, 2022.