令和4（2022）年度　卒業研究報告書概要

課程，学籍番号，氏名	課程: 電気・電子情報工学課程，学籍番号: B213258，氏名: 樋口 拓真
工学分野名: 情報通信システム	指導教員名: 市川 周一

題　　目: 和　ソフトプロセッサCometにおける専用命令拡張方法

（英　Special Instruction Extension Method in Soft Processor Comet　）

Abstract

Comet is an open-source soft core processor of RISC-V architecture. Comet was adopted in a preceding study, where a gsm special instruction was implemented in high-level synthesis using Xilinx Vivado HLS.

This research adopts Vitis HLS instead of Vivado HLS, and compares the new implementation with the preceding implementation. This study also examines the feasibility of sha special instruction for CHStone benchmark.

The RESOURCE pragma, that is no longer supported by Vitis HLS, was replaced by the BIND_STORAGE pragma. In a follow-up examination, the combination of the PIPELINE and INLINE pragmas was the most effective. In the implementation of the sha special instruction, C Simulation was successful. However, the estimations of latency and hardware resource were large, which caused errors in Co-Simulation.

概　　要

ソフトプロセッサとは，ハードウェア記述言語 (HDL) で記述され，論理合成で実装することの出来るマイクロプロセッサである．また，高級言語による動作記述から, HDL の記述を自動的に生成する技術を高位合成 (HLS) と呼ぶ．高位合成ツールの進歩を踏まえて，Skalicky ら (2015) や Rokicki ら (2019) は，C 言語や C++ 言語で記述したソフトプロセッサを発表している．

坂本 (2017) は，Skalicky らの手法を基に高位合成可能な MIPS の命令セットシミュレータ SPIM-like を自作し，CHStone ベンチマークのアプリケーションから一部の関数を専用命令として実装した．岩本 (2018) は，実装するアプリケーションを拡大し，関数の選択に実行時間優先とコードサイズ優先という 2 つの基準を設定した．板東ら (2022) は，商用アーキテクチャである MIPS を研究で使用する際の制約を考慮し，オープンソースな RISC-V アーキテクチャを採用したソフトプロセッサ Comet を評価基盤として使用した．Comet に対して実装可能な CHStone ベンチマークのアプリケーションを 7 種類とし，gsm を専用命令として実装したが，最新ではない高位合成ツールの Vivado HLS を用いていた．

本研究では，Vivado HLS の後継である Vitis HLS を用いて板東らの追試を行い，Vitis HLS への移行に伴う各種不具合に対応する．また，板東らにより実装可能とされている CHStone ベンチマークのアプリケーションうち，実装されていない sha について実装方法を検討する．

移行に伴う不具合への対応として，Vitis HLS で今後サポート外となる RESOURCE プラグマを BIND_STORAGE プラグマへ置換し，同様の合成結果が得られることを確認した．

追試では，gsm における関数の引数を値渡しで受け渡し，専用命令として実装した．実行時間の指標には Latency，ハードウェアリソース使用量の指標には Slice 数を使用した．合成結果を先行研究と比較すると，Vitis HLS で高位合成を行うことでハードウェアリソース使用量の改善が見られた．また，PIPELINE プラグマ，INLINE プラグマの有無による合成結果への影響について評価を行った．実行時間優先の場合もコードサイズ優先の場合も PIPELINE プラグマ，INLINE プラグマの併用が最も効果的であった．

sha 専用命令の実装では，実行時間優先，コードサイズ優先で同一の関数が選ばれた．sha の関数はグローバル変数を操作する関数であるため，グローバル変数のアドレスを参照渡しするように修正した．C Simulation には成功したが，その後に行った高位合成では Latency およびハードウェアリソースの推定値が非常に大きくなり，Co-Simulation でエラーが発生している．

Comet に対して，Vitis HLS を用いた高位合成で専用命令の実装および評価を行う環境を確立出来たが，sha 専用命令の実装については不具合が生じているため，不具合の修正を行う必要がある．

発表する際の課程を記入　電気・電子情報工学　課程　発表番号 75

（学籍が他課程所属の学生も発表する課程を記入すること）