

令和3（2021）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：B203256, 氏名：板東伸彦	
工学分野名：情報通信システム	指導教員名：市川周一	
<p>題 目：ソフトプロセッサ Comet における専用命令拡張方法の検討</p> <p>(A Study on Special Instruction Extension Method for Soft Processor Comet)</p>		
<p>Abstract</p> <p>Though our previous studies adopted MIPS instruction set as evaluation platform, commercial architectures may impose limitations on their use in research. In this study, we adopt Comet, a soft processor of RISC-V architecture, which is an open standard architecture. The first purpose of this study is to apply and evaluate HLS directives to Comet. The second purpose is to implement special instructions to Comet. As a result of the first objective, we found that the best directives for Comet are “pragma HLS pipeline” and “pragma HLS inline”. For the second purpose, we implemented a special instruction specifying those directives and compared it with previous studies.</p>		
<p>概 要</p> <p>組込みシステムでは、ハードウェア記述言語 (HDL) で記述されたソフトプロセッサが多く利用されている。一方近年では、C 言語などの高級言語による動作レベルの記述から、HDL 記述を自動的に生成する高位合成 (HLS) 技術が提供されている。</p> <p>Skalicky ら(2015)は C 言語で記述された命令セットシミュレータからソフトプロセッサを合成する研究を行った。坂本ら(2018)はソフトウェアの一部の関数を専用命令として組み込んだプロセッサを生成することを提案した。Skalicky らの手法をもとに、MIPS の命令セットシミュレータである SPIM-like を自作して研究した。岩本ら(2019)は坂本の手法に基づき、専用命令の対象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、評価した。正信(2020)は岩本の提案したプロセッサに HLS ツールのディレクティブを適用し、パフォーマンスの向上などによる最適化を提案した。</p> <p>以上の研究では、命令セットとして MIPS アーキテクチャを採用していた。しかし商用アーキテクチャでは、研究上の使用に制約が生じる場合がある。そこで本研究では、オープン標準である RISC-V アーキテクチャのソフトプロセッサ Comet を用いる。本研究の目的は 2 つあり、1 つめは Comet にディレクティブを適用して評価することである。2 つめは Comet に専用命令を追加する手法を検討することである。</p> <p>最初の目的であるディレクティブの適用については、pragma HLS resource, pragma HLS inline, pragma HLS pipeline, pragma HLS unroll を選択した。ハードウェアリソースの指標にはスライスを、実行時間の指標にはレイテンシを用いた。結果は pragma HLS unroll 以外は実装に成功した。ディレクティブが無い場合と本研究で実装に成功した全てのディレクティブがある場合では、スライスは約 23.3%、レイテンシは約 57.8% 削減された。また、Comet に指定するディレクティブとして pragma HLS pipeline と pragma HLS inline には相乗効果があることが分かった。</p> <p>2 つめの目的である専用命令の追加に関しては、先行研究と同じく CHStone ベンチマークを対象とした。Comet の仕様や RISC-V 用のクロスコンパイラの制限により実装不能なアプリケーション以外で、先行研究と同じ条件で実装が可能な gsm を実装し、評価を行った。コードサイズ優先の場合、先行研究と比べてスライスが約 10.8%削減され、レイテンシは約 286.3%増加した。実行時間優先の場合、先行研究と比べてスライスが約 15.6%削減され、レイテンシは約 130.4%増加した。以上の結果より Comet はハードウェアリソースの面で優れており、SPIM-like はレイテンシの面で優れていることが分かった。今回は gsm だけを実装したが、CHStone ベンチマークの他のアプリケーションの実装評価は今後の課題とする。</p>		

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

63

(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)