

電気・電子情報工学専攻	学籍番号	M143231	指導教員氏名	市川 周一 藤枝 直輝
申請者氏名	坂本 譲二			

論 文 要 旨 (修 士)

論文題目	プロセッサの高位合成および特殊命令の実装・評価
------	-------------------------

ソフトウェアの処理手順など知的財産権の保護は重要な課題である。保護手法の1つとして処理の一部のハードウェア化がある。一方、組込みシステムではハードウェア記述言語 (HDL) によって合成可能なソフトプロセッサが広く利用される。

Skalicky らは高位合成を用いてプロセッサシミュレータからソフトプロセッサを生成する手法を提案した。Skalicky らの手法ではアプリケーションで使用されない命令をソフトプロセッサから除外することでハードウェア使用量を削減している。しかし、実用的なベンチマークを用いた検証はなされておらず、ハードウェア使用量についても同じ命令セットを実装する軽量なソフトプロセッサと比較していない。

本研究では、まず先行研究の追加検証を行うことを目的とする。次に先行研究の手法で生成するソフトプロセッサに特殊命令を実装する手法を提案する。

追加検証ではアプリケーションとして CHStone ベンチマークを用い、オープンソースの MIPS 互換プロセッサ Plasma とハードウェア使用量を比較した。各ベンチマーク単体では、blowfish が最も SLICE 数が少なく、Plasma と比較して 40.4% 小さい。最も SLICE 数が多いものは aes で Plasma より 21.2% 大きい。Plasma より SLICE 数が多くなったものは 12 個の内 aes だけだった。ベンチマークを複数実行する場合、4 個の場合でも最も命令数が少ない組み合わせでは SLICE 数が 452 で Plasma より小さい。つまり複数実行する場合でも組み合わせ次第では先行研究の手法は有効であると言える。

特殊命令の実装は C 言語の関数単位で行った。アプリケーション側の関数をシミュレータに組み込んで高位合成を行うことでハードウェア化した。特殊命令化した関数へ引数を渡す場合、値渡しでは MIPS の仕様に従いレジスタおよびスタックを用いて値を渡した。参照渡しでは 3 つの手法を提案した。1 つ目はメインメモリのアドレスを渡す手法、2 つ目はスクラッチパッドメモリ (SPM) を利用する手法、3 つ目は一時変数を利用する手法である。CHStone ベンチマークの adpcm, aes, motion の 3 つのアプリケーションで特殊命令を実装し、ハードウェア使用量および実行サイクル数を検証した。adpcm では 4 つの関数を特殊命令化した。関数は全て値渡しの引数しか持たない。合成の結果、特殊命令の実装によって SLICE 数が 20.4% 増加し、実行サイクル数が 31.7% 減少した。SLICE 数と実行サイクル数の積 (AT 積) で評価した場合、特殊命令を実装した方が 17.8% AT 積が小さい。特殊命令化によってセキュリティだけでなく AT 積の向上も期待できることが分かった。aes では 3 つの関数を特殊命令化し、そのうち 2 つは参照渡しの引数を持つ。参照渡しの 3 つの実装手法の中では一時変数を利用するものが最も AT 積が小さかった。しかし、特殊命令を実装してないものと比べると 67.3% AT 積が大きい。参照渡しの実装はハードウェアが大きくなる場合があるためである。motion では参照渡しの引数を持つ 1 つの関数を特殊命令化した。型変換の問題で一時変数を利用する手法以外では実装エラーとなった。AT 積は特殊命令を実装しないものと比べて 78.0% 小さい。参照渡しの実装手法については一時変数を利用するものが性能も適応性も高いことが分かった。