C-015

高位合成による専用命令実装手法の再検討
The HLS implementation of special instructions: revisited

岩原 和輝‡ 正信 健人‡ 市川周一‡
Kazuki Iwahara Kento Masanobu Shuichi Ichikawa

1. はじめに

組込みシステムではハードウェア記述言語で記述されたソフトプロセッサが多く利用されている．

近年では，高位合成 (HLS)ツールの進歩を踏まえて，C 言語で記述されたソフトプロセッサも発表されている．Ahmed ら[1]は HLS ツールを利用し，C 言語による記述からソフトプロセッサを合成した．Skalicky ら[2]は命令セットシミュレータからソフトプロセッサを合成した．これらの研究では，HLS ツールのディレクティブを利用し,スループットやパフォーマンスの向上などが期待されている[2][3]．ただし，各々のディレクティブに対する最適化の定量的な評価は行われていない．

岩本ら[4]は，Skalicky ら[2]の手法をベースに，アプリケーションの処理の一部を専用命令として実装し，命令セットシミュレータを高位合成してソフトプロセッサを生成する手法を提案した．評価には，CHStone[5]ベンチマークの内のアプリケーション 11 種類を使用した．また，専用命令化する関数は実行時間, コードサイズといった明確な基準で選択した．そして，手法適用時に発生した問題について報告した．

本研究では，岩本ら[4]の実装の追実験を行い，手法適用時の問題と対策を明確化することを目的とする．また，追加検証として，HLS ツールのディレクティブを指定することによるソフトプロセッサ性能・回路規模の変化を検証する．

2. 実験方法

2.1 高位合成を用いたソフトプロセッサ Spim-like

Spim-like は，岩本ら[4]が使用したソフトプロセッサである．Skalicky ら[2] の手法をベースに命令セットシミュレータを高位合成している．アプリケーションの一部の関数をシミュレータ側に移すことで専用命令として実装している．

2.2 実験環境

実験環境として，仮想 OS 上にクロスコンパイル環境を構築した．構築したクロスコンパイル環境の詳細を表 1 に示した．Spim-like のシミュレータ記述を Xilinx 社の Vivado HLS により高位合成し，得られたハードウェア記述を Vivado により論理合成した．高位合成および論理合成の開発環境の詳細を表 2 に示した．

Table 1.アプリケーションのクロスコンパイル環境

VM Software	Oracle VMVirtualBox6.1.14
Guest OS	Ubuntu 18.04.4
Compiler	mipsel-linux-elf-gcc ver5.4.0
Architecture	MIP32 Release 1
Benchmark Software	CHStone v1.11

Table 2.高位合成および論理合成の開発環境

HLS Tool	Vivado HLS 2020.1
Logic Synthesis Tool	Vivado 2020.1
Target Device	Artrix-7 xc7a100tcsg324-1

2.3 問題点の明確化

Vivado HLS 上で C シミュレーションを行った後，Spim-like で CHStone ベンチマークをソフトウェアとして実行した．そして，出力されたセルフチェック値から計算結果の一致を確認した．

jpeg において C シミュレーションエラーが発生した．この問題は，未解決につき，今後の課題とした．

専用命令を実装した Spim-like_sp では，専用命令において複数の参照渡しを用いた．また，専用命令として実装する関数の選択基準を複数設けた．参照渡しの種類および関数の選択基準を表 3，4 に示す．

Table 3. 専用命令における参照渡しの種類

sp	参照渡しが不要な実装
ptr	メモリのポインタを直接渡す参照渡し
spm_s	SPM を用いたソフトウェアにおける参照渡し
spm_h	SPM を用いたハードウェアにおける参照渡し

Table 4. 専用命令として実装する関数の選択基準

excution time	最も実行時間を要している関数
code size	最もコードサイズが大きい関数
time/size	優先度が実行時間/コードサイズで同一

全 39 種の実装パターンの内, 計算結果の一致を確認できたのは 16 種である．adpcm (spm_s) および, jpeg のコードサイズ優先の全ての実装において C シミュレーションエラーが発生した．また，motion (ptr)にて，高位合成エラーが発生した．ただし，C シミュレーションでの計算結果は正しかった．

Spim-like_sp において, motion (ptr) を除く計算結果の一致を確認した．これら 15 種の実装を, より複雑な実装のプロセッサとしてディレクティブを指定した検証の対象とした．

‡ 豊橋技術科学大学　Toyohashi University of Tecfnology

3. ディレクティブの指定と評価

追実験を行った Spim-like_sp に対して，ディレクティブ「pragma HLS pipeline」を専用命令として追加した関数に指定した．

測定結果は CHStone ベンチマーク 11 種のアプリケーションにおける平均値を,ディレクティブなしの場合を基準とした相対値で示した．評価基準とするため，ディレクティブを指定しない場合の測定を事前に行っている．まず，高位合成した Spim-like_sp を用いて協調シミュレーションを行い，実行サイクル数（レイテンシ）を指標として，実行時間の変化を測定した．次に，高位合成によって生成されたハードウェア記述を Vivado で論理合成した際に出力される論理ブロック数（SLICE 数） を指標として，ハードウェアリソース使用量の変化を測定した．最後に，Slice 数とレイテンシの積から算出した AT 積を指標として，トレードオフの評価を行った．

Spim-like_sp に pragma HLS pipeline を指定した場合の実行時間・ハードウェアリソース使用量の評価は以下のとおりである．レイテンシ・ Slice 数のディレクティブなしの場合に対する相対値を図 1 と図 2 に示した．

専用命令化された関数が呼び出す関数 2 つを含む計 3 つの関数がシミュレータ側に移されている．それぞれの関数にディレクティブを指定したことで，他の実装に比べディレクティブの指定個所が多くなったことが削減率の大きい要因であると考えられる．また，他の実装に比べ，アプリケーションの処理全体のレイテンシが少ないため，相対的に削減率が大きいと考えられる．

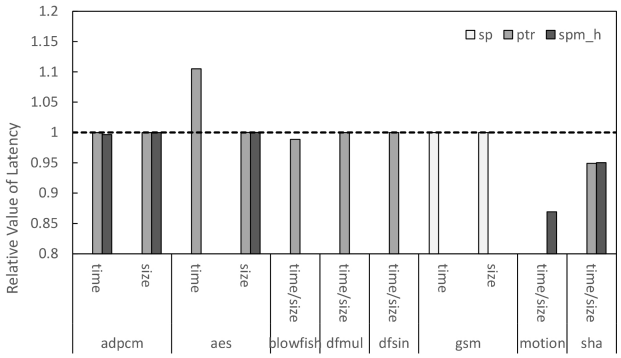

図 1 ディレクティブを指定した Spim-like_sp のレイテンシの相対値 (pragma HLS pipeline)

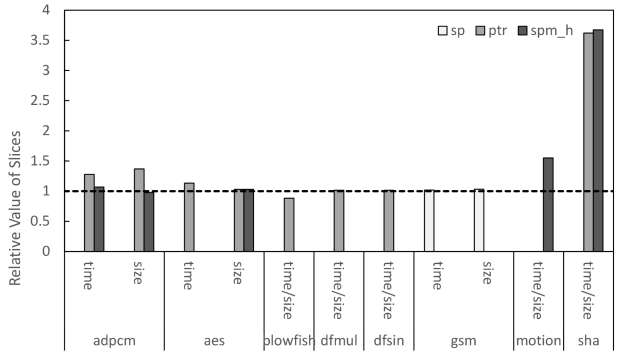

図 2 ディレクティブを指定した Spim-like_sp の Slice 数の相対値 (pragma HLS pipeline)

トレードオフの評価として，ディレクティブなしの場合に対する AT 積の相対値を図 3 に示した．

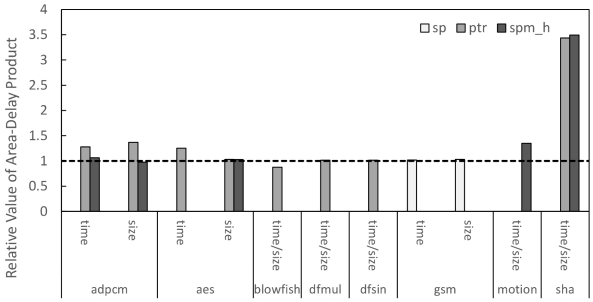

図 3 ディレクティブを指定した Spim-like_sp の AT 積の相対値 (pragma HLS pipeline)

4. おわりに

本研究における未解決問題として，adpcm (spm_s)および, jpeg のコードサイズ優先の全ての実装で C シミュレーションエラーが発生したことが挙げられる．また，motion (ptr) で高位合成エラーが発生したことがある．これらの不具合の修正を今後の課題とする．

謝辞

本研究の一部は JSPS 科研費 20K11733 によるものである．

参考文献

[1] T. Ahmed, N. Sakamoto, J. Anderson, and Y. Hara-Azumi, “Synthesizable-from-C Embedded Processor Based on MIPS-ISA and OISC” Proceedings of IEEE 13th International Conference on Embedded and Ubiquitous Computing (EUC 2015), pp.114-123, Oct. 2015.

[2] S. Skalicky, T. Ananthanarayana, S. Lopez, M. Lukowiak, “Designing customized ISA processors using high level synthesis,”2015 International Conference on ReConFigurable Computing and FPGAs, pp.1-6 ,Dec. 2015.

[3] P. Mantovani, et al., “HL5: A 32-bit RISC-V Processor Designed with High-Level Synthesis,”2020 IEEE Custom Integrated Circuits Conference, pp.1-8,2020.

[4] 岩本凌大,藤枝直輝,市川周一,坂本譲二,”高位合成による専用命令実装手法の予備的評価”,電子情報通信学会技術報告, vol.118,no.432, pp.101-106, 2019.

[5] Y. Hara, H. Tomiyama, S. Honda, H. Takada, “Proposal and Quantitative Analysis of the CHStone Benchmark Program Suite for Practical C-based High-level Synthesis,” Journal of Information Processing, vol.17, pp.242-254, 2009.