様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

令和　２ 年　 ６ 月 １７ 日現在

機関番号：３２６７８
研究種目：基盤研究(C)（一般）
研究期間：2016～2019
課題番号：１６Ｋ００００８４

研究課題名（和文）高位合成による集積回路設計におけるメモリアクセス自動最適化に関する研究

研究課題名（英文）Memory access optimizations for VLSI design with high-level synthesis

研究代表者
瀬戸　謙修（SETO, Kenshu）

東京都市大学・理工学部・講師

研究者番号：１０４２０２４１

交付決定額（研究期間全体）：（直接経費）　　3,600,000円

研究成果の概要（和文）：高位合成技術は、ハードウェアの設計期間を劇的に短縮するが、入力Cプログラムの書き換えが必要となる場合がある。メモリアクセスを最適化するための書き換えについて、従来技術の制限を取り除く手法を提案、自動化ツールを開発した結果、従来技術と比べて、2.8倍の性能向上を得るとともに、回路面積を35%削減できた。本研究成果により３本の査読有り論文を出版するとともに、情報処理学会TSLDM Best paperを受賞した。

研究成果の学術的意義や社会的意義
ソフトウェアを自動でハードウェア化する高位合成技術が、性能向上、低消費電力化のために活用されているが、画像処理や行列演算などメモリアクセスが頻繁なソフトウェアは、そのまま高位合成しても性能向上が難しく、ソフトウェアの人手最適化を行うために長時間の手間が必要となっていた。本研究では、ソフトウェア中のメモリアクセス自動最適化技術の開発に成功した。この結果、開発者がソフトウェアのハードウェア化によるメリットを享受しやすくなる。

研究成果の概要（英文）：High-level synthesis significantly reduces the hardware design time, however, manual rewriting of C code is often necessary. In this research, we proposed automatic memory access optimization techniques that push the envelope of the previous techniques. According to the experimental results, the proposed method achieved 2.8x performance enhancement with 35% chip area reduction. We published 3 reviewed journal papers, one of which received the IPSJ TSLDM best paper award.

研究分野：高位合成、並列化コンパイラ、ハードウェア・ソフトウェア協調設計、VLSI設計技術

キーワード：高位合成　メモリアクセス最適化　スカラリプレイス

様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９（共通）

１．研究開始当初の背景

C 言語などのプログラムから、アクセラレータなどのハードウェア設計(Verilog-HDL 記述)を自動生成する高位合成は、集積回路の短期設計に欠かせない技術である。しかしながら、複雑なプログラムから性能要求を満たすハードウェア設計を出力するには、高位合成適用前に、アーキテクチャ、高位合成、およびコンパイラ最適化技術を意識したプログラム書き換え作業が必要となる。この書き換えには、十分な専門知識が必要であり、かつ、時間を浪費する試行錯誤が必要となるため、高位合成の幅広い活用を妨げている。高位合成による高速化が求められる処理として、画像フィルタ処理や行列演算など、多重ループ中に多数の配列アクセスが含まれるプログラムが挙げられる。多数の配列アクセスは多数のメモリアクセスに対応し、性能のボトルネックとなるため、通常、配列アクセス部分の書き換えによる、メモリアクセスの高速化が必要となる。書き換え自動化を目的として、アフィンプログラム（ループ範囲、if 文条件、配列の添え字がループ変数の線形式であるプログラム）を対象とした、メモリアクセス最適化の研究が進んでいる。アフィンプログラム向けの代表的なメモリアクセス最適化技術として、(1)メモリ分割技術、(2)スカラリプレイス技術、の２つが挙げられる。

(1) メモリ分割は、アクセスの集中する配列(ローカルメモリ)を分割し(分割されたメモリをバンクと呼ぶ)、複数アクセスを同時処理することで、ハードウェア高性能化に効果的である。メモリ分割では、バンクからアクセス先に適切なデータを送るためのマルチプレクサが必要となる。

(2) スカラリプレイスは、ローカルメモリからアクセスした配列要素を、シフトレジスタに格納し、後のアクセスでは、ローカルメモリではなくシフトレジスタをアクセスすることで、メモリへのアクセス混雑を防ぐ。スカラリプレイスは一時配列削除に有効である。

(1)(2)のメモリアクセス最適化は、適用対象となる配列アクセスパターンやプログラム記述に制約があるため、実際のプログラムには適用できない場合や効果が限定的な場合が少なくない点が課題であった。

２．研究の目的

本研究では、プログラムのソースコードに対して適用するメモリアクセス最適化について、現状のソースコードに対する制約を取り除く研究を進め、ツールを開発し、自動化を行うことを目的とする。これによりソフトウェアを迅速にハードウェア化し、アプリケーションの性能向上および低消費電力化を行うことが期待できる。ハードウェア化によるメリットが得られる処理はループであることが多いため、ループを対象としたメモリアクセス最適化を対象とする。C 言語では、配列変数へのアクセスが、メモリアクセスに対応する。したがって、配列変数へのアクセスをなんらかの方法で削減できれば、メモリアクセスを削減することができる。スカラリプレイスでは、配列アクセスによって得られたデータをスカラ変数などの一次変数に保存し、次回以降の同じ配列要素へのアクセスに対しては、配列ではなく保存したスカラ変数を再利用することで、配列アクセス、したがって、メモリアクセスを削減する技術である。スカラリプレイスでは、スカラ変数をシフトレジスタの形式で構成することで、ループをまたいだ再利用を可能としている。なお、本研究では、C プログラムを対象とするが、画像処理向けのドメイン特化型言語である Halide や、Python などの抽象度の高い言語においても、本研究で提案した手法は拡張可能である。スカラリプレイスはメモリアクセス最適化として効果的だが、以下のような限界があり、実用化の障壁となっている。本研究では、これらの課題を解決することを目的とした。

(1) 配列変数へのアクセスの際、配列アクセスの添え字に定数が含まれる場合を扱うことができない。例えば、画像のフィルタ処理において画像の端を処理する際、配列アクセスの添え字に定数が含まれる場合があるため、スカラリプレイスを適用することができない。

(2) スカラリプレイスによるメモリアクセス最適化を行った結果、長いシフトレジスタが発生するため、回路面積が大幅に増大してしまう。特に、大規模画像などを扱う処理において、スカラ変数に保存したデータを再利用するまでに回るループ回数が多い場合に顕著となる。

(3) 対象とするループのまわる回数が定数ではなくコンパイル時に決まる記号定数の場合にスカラリプレイスを適用すると、シフトレジスタに多入力のマルチプレクサを追加する必要が発生するため、回路規模が増大する。

(4) スカラリプレイスを適用後、シフトレジスタに配列からアクセスしたデータを初期化データとして設定する必要がある。その際に、一つの配列アクセスではなく、複数の配列アクセスからのデータを設定する必要がある場合、ループピーリングと呼ばれるループ変換を適用することで内側ループから配列アクセスを削減する後処理を行っていたが、コード量の増大を伴うため回路規模が増大し、並列実行部分の減少により性能向上が限定的となる。

３．研究の方法

対象とする C コードは、Static Control Part (SCoP)と呼ばれる、C 言語のサブセットで記述されるものとした。SCoP では、if 文の条件や、配列アクセスの添え字や、for ループのループ

回数の上限が、ループ変数の線形式に限定されたプログラムである。C コードを SCoP と呼ばれるサブセットに制限しても、多くの画像処理や行列演算を扱うことができるとともに、データ再利用解析など、メモリアクセス最適化に必要な静的な解析が行いやすくなる利点がある。スカラリプレイスについてはさらに C コードがステンシル計算であるものを対象した。

ループ中の配列アクセスには、静的なものと、動的なものに分類できる。C コード中の静的な配列アクセスはプログラムテキスト中に出現する配列アクセスのことで、動的なアクセスとは C コードの実行中に発生する実際の各配列アクセスのインスタンスのことである。本研究で削減対象とする配列アクセスは、静的な配列とした。なぜなら、高位合成によるループパイプライン化による性能向上のためには、静的な配列アクセスの削減が重要なためである。したがって、以降で配列アクセスとは、静的な配列アクセスのことを指す。スカラリプレイスでは、ループ中の同一配列への複数のアクセスを削除することを試みる。具体的には、最初に配列の各要素へアクセスする配列アクセス１つ（再利用元と呼ぶ）だけを残して、残り（再利用先と呼ぶ）をすべて削除することを試みる。そのためには、各配列アクセスが、アドレス空間のどのデータをアクセスするか、また、ある配列アクセス(再利用元)がアクセスしたデータを、別の配列アクセス(再利用先)がのちにアクセスするか、などを厳密に解析する必要がある。

このような解析を行うため、SCoP で表現された多重ループに対して適用可能な解析技術である、多面体モデルと呼ばれる技術を使用した。多面体モデルは主にループプログラムの並列化コンパイラの基本技術として活用されているが、メモリアクセス最適化にも活用できる。多面体モデルではループ変数 i，j，k，…の値を並べたベクトルを、繰り返しベクトルと呼び、繰り返しベクトルを使用して、(1) ループ中の各実行文（配列アクセス）が実行される範囲、(2) 配列アクセスのアクセスするアドレス空間上のアドレス、(3) 各実行文（配列アクセス）の実行順序、を表現する技術である。プログラム中の各配列アクセスを多面体モデルで表現することで、各配列間でのデータ再利用の関係や、再利用される条件を、ループ変数を使用した形で表現できる。その際、プログラム中の各配列を多面体モデルで表現した後、多面体モデルに対する様々な操作を行うことが可能な C ライブラリである、Integer Set Library (ISL)を活用した。具体的には、ISL は、格子点の集合や、格子点間の写像に対して、様々な処理を行うことを通して、多面体モデルの基づく配列アクセスの解析に用いることができる。例えば、本研究では、ISL を使用して、配列アクセス u から配列アクセス v へ、再利用が発生する際の繰り返しベクトル iu，iv の関係式（再利用関係）を求め、再利用関係から、再利用ベクトルや、再利用距離と呼ぶ量を求めた。再利用距離は、配列アクセスをシフトレジスタへのアクセスに置き換える際に必要となる情報で、再利用距離に対応する位置にあるシフトレジスタに置き換えた。

研究の目的に挙げた(1)～(4)の課題を解決するため、以下の項目を目的とした。(1) 添え字に定数を含んだ再利用先の配列アクセスをスカラリプレイスする際に、参照するシフトレジスタの位置が繰り返しベクトルに応じて異なることに注意して、再利用先の実行条件に応じて参照するシフトレジスタの位置を切り替えるアルゴリズムを考案し、ISL を使用して、ツールを作成することを目指した。(2) また、シフトレジスタをアクセスする位置を境界として部分シフトレジスタに分割し、長い部分シフトレジスタをデュアルポートメモリで実現する循環バッファとして実現することで、シフトレジスタの回路規模を抑えることを目指した。(3) 循環バッファを用いることで、ループ回数が記号定数の場合にも、記号定数の最大値を与えることで、スカラリプレイスを適用することを目指した。(4) シフトレジスタに複数の配列からアクセスしたデータを設定する必要がある場合に、ループ範囲を拡張するとともに配列アクセスを新規に追加することで、ループピーリングを避ける方法の考案を目指した。

４．研究成果

研究の目的に挙げた(1)～(4)の課題を解決することに成功した。その目的を達成するため、ISL(Integer Set Library)を使用して、自動スカラリプレイスツールを作成した。このツールは、今後のさらなる研究の基盤ソフトウェアとして用いることができる。配列の添え字が定数の場合に対応することができる技術を研究開発し、国際会議１件、論文誌１件で発表した。この技術を用いることで、従来技術と比べて、2.8 倍の性能向上を得るとともに、回路面積を 35%削減できている。このツールを拡張し、シフトレジスタが長い場合に循環バッファとして実装する技術や、ループ回数がパラメータの場合を扱う技術もツール実装し、回路の省面積化に効果があることを確認した結果を論文誌１件として発表した。この成果により、回路性能を維持しながら、回路面積を 50%削減できた。さらに、スカラリプレイス前に、前処理のコード変換を行うことで、コードサイズ増大のオーバーヘッドを伴うループピーリングを行わずに、シフトレジスタの初期化に必要な配列アクセスを削減する方法を提案し、ツールに実装した。提案手法を評価した結果、生成されたハードウェアにおいて、最大で、40%の実行時間削減、27%の面積削減を達成できた。この成果は、論文誌１件で発表した。開発したメモリ最適化技術を適用したニューラルネットワークアクセラレータについても、国際会議で１件発表を行った。

従来のスカラリプレイスでは、外部メモリからローカルメモリにループ処理に必要となるデータをすべてコピーした後に、スカラリプレイスを適用したコードを実行する方式としていたが、ローカルメモリサイズが増大するだけでなく、コピー時間のオーバーヘッドによりサイクル数

が増大することが問題となっていた。このようなスカラリプレイスの問題に対し、外部メモリからローカルメモリへのデータコピーがすべて完了することを待つことなく、データ転送と計算の同時実行を行うための C コード最適化方式を提案した。高位合成および論理合成を実行して評価を行った結果、提案手法は、従来手法と比べ、動作周波数を向上させ、ハードウェア面積を削減できた。さらに、ループの次元数が配列の次元数よりも大きい場合、具体的には例えば行列積計算に出現するような３次元ループ中の２次元配列に対するスカラリプレイスにおいて、複数の入力配列が存在する場合に、どの順番で入力配列をコピーすると性能的および面積的に効率が良いかを、自己再利用距離に基づいて決定する方法を提案した。提案手法を適用することで、ハードウェア実行に必要なサイクル数、および、回路面積について、従来のスカラリプレイス技術と比較して、それぞれ最大 39.0%、12.5%削減できた。この成果については、国内会議で２件の発表を行った。また、この技術を AD 変換器内部のデジタル回路に適用し、処理性能を向上できた。

メモリ分割技術についても、配列のアドレス空間を異なるバンクに割り当てるマッピング関数およびバンク内のアドレスを決定するオフセット関数を算出し、最適化後のコードを出力するツールを開発した。また、グラフ彩色を用いてメモリ分割を行った結果のバンク割り当てとオフセット決定をルックアップテーブルで計算する方法が、2017 年にコーネル大学の研究グループから提案されていたため、そのアルゴリズムをツール実装し、複雑な添え字を持つ配列アクセスに対しても、メモリ分割を行えることを確認した。また、アドレス計算部分の面積削減のため、ルックアップテーブルではなく、算術式によるバンク割り当てとオフセット決定手法についても検討を進めた。また、メモリ分割技術と、スカラリプレイスとメモリ分割を組合せる方法を考案し、国際会議で１件発表した。

５．主な発表論文等

〔雑誌論文〕　計4件（うち査読付論文　4件／うち国際共著　0件／うちオープンアクセス　3件）

１．著者名 Kenshu Seto	４．巻 13
２．論文標題 Shift Register Initialization in Scalar Replacement for Reducing Code Size	５．発行年 2020年
３．雑誌名 IPSJ Transactions on System LSI Design Methodology	６．最初と最後の頁 2～9
掲載論文のDOI（デジタルオブジェクト識別子） https://doi.org/10.2197/ipsjtsldm.13.2	査読の有無 有
オープンアクセス オープンアクセスとしている（また、その予定である）	国際共著 －

１．著者名 Kenshu Seto	４．巻 12
２．論文標題 Scalar Replacement with Circular Buffers	５．発行年 2019年
３．雑誌名 IPSJ Transactions on System LSI Design Methodology	６．最初と最後の頁 13～21
掲載論文のDOI（デジタルオブジェクト識別子） https://doi.org/10.2197/ipsjtsldm.12.13	査読の有無 有
オープンアクセス オープンアクセスとしている（また、その予定である）	国際共著 －

１．著者名 Yuji Shindo, Kenshu Seto, Hao San	４．巻 139
２．論文標題 Area Reduction Technique for Digital Circuit Part in Non-Binary Analog-to-Digital Converter	５．発行年 2019年
３．雑誌名 IEEJ Transactions on Electronics, Information and Systems	６．最初と最後の頁 76～82
掲載論文のDOI（デジタルオブジェクト識別子） https://doi.org/10.1541/ieejeiss.139.76	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 －

１．著者名 Kenshu Seto	４．巻 12
２．論文標題 Scalar Replacement with Polyhedral Model	５．発行年 2018年
３．雑誌名 IPSJ Transactions on System LSI Design Methodology	６．最初と最後の頁 -
掲載論文のDOI（デジタルオブジェクト識別子） https://doi.org/10.2197/ipsjtsldm.12.1	査読の有無 有
オープンアクセス オープンアクセスとしている（また、その予定である）	国際共著 －

〔学会発表〕　計7件（うち招待講演　1件／うち国際学会　3件）

1．発表者名
Yuji Toda, Kenshu Seto

2．発表標題
Simultaneous Application of Memory Partitioning and Scalar Replacement

3．学会等名
TJCAS 2019（国際学会）

4．発表年
2020年

1．発表者名
Kenshu Seto, Hamid Nejatollahi, Jiyoung A, Sujin Kang, Nikil Dutt

2．発表標題
Small Memory Footprint Neural Network Accelerators

3．学会等名
International Symposium on Quality Electronic Design (ISQED)（招待講演）（国際学会）

4．発表年
2019年

1．発表者名
Kenshu Seto

2．発表標題
Scalar Replacement with Array Dataflow Analysis for Hardware Synthesis

3．学会等名
Forum on specification & Design Languages (FDL) 2017（国際学会）

4．発表年
2017年

1．発表者名
進藤 佑司, 瀬戸 謙修, 傘 昊

2．発表標題
β展開に基づくAD変換器のルックアップテーブル除去によるデジタル回路部の面積削減

3．学会等名
デザインガイア2017

4．発表年
2017年

１．発表者名
石川 大輔，瀬戸 謙修

２．発表標題
高位合成における多重ループに対するパイプライン処理時のサイクル数オーバーヘッド削減を行うループ平坦化ツールの開発

３．学会等名
電子情報通信学会 VLD研究会

４．発表年
2018年

１．発表者名
石川 大輔，瀬戸 謙修

２．発表標題
計算と通信を同時に行うハードウェアを生成するメモリアクセス最適化技術

３．学会等名
第29回 回路とシステムワークショップ

４．発表年
2016年

１．発表者名
石川 大輔，瀬戸 謙修

２．発表標題
高位合成によるアクセラレータ設計を対象としたサイクル数削減およびバッファサイズ最小化のためのデータ転送最適化手法

３．学会等名
デザインガイア2016

４．発表年
2016年

〔図書〕　計0件

〔産業財産権〕

〔その他〕

―

６．研究組織

	氏名 （ローマ字氏名） （研究者番号）	所属研究機関・部局・職 （機関番号）	備考