

平成 22 年 6 月 1 日現在

研究種目： 基盤研究 (C)  
研究期間： 2007年度～ 2009年度  
課題番号： 19500031  
研究課題名 (和文) 高位合成向け並列化コンパイラ共通基盤の研究  
研究課題名 (英文) Parallelizing Compiler Infrastructure for High-level Synthesis  
研究代表者  
滝本 宗宏 (TAKIMOTO MUNEHIRO)  
東京理科大学・理工学部・講師  
研究者番号：00318205

## 研究成果の概要 (和文)：

既成のコンパイラ・インフラストラクチャ COINS を改良して、ソフトウェアから専用のハードウェアを生成する高位合成向けのインフラストラクチャを作成した。本インフラストラクチャによって、従来のコンパイラ技術と高位合成技術を融合し、各システムに最適なソフトウェアとハードウェアの協調計算を実現できる。

## 研究成果の概要 (英文)：

Improving the compiler infrastructure which is called COINS, we have implemented a new infrastructure for high-level synthesis which produces the specific hardware for software. The infrastructure enables optimally cooperative computing between software and hardware for each system.

## 交付決定額

(金額単位：円)

	直接経費	間接経費	合計
19年度	2,100,000	630,000	2,730,000
20年度	800,000	240,000	1,040,000
21年度	600,000	180,000	780,000
年度			
年度			
総計			

研究分野：総合領域

科研費の分科・細目：情報学・ソフトウェア

キーワード：高位合成, コンパイラ, FPGA, コード最適化, コンパイラ共通基盤, スケジューリング, 並列化, 投機的実行

## 1. 研究開始当初の背景

システム・オン・チップの進歩によって、振舞い記述からハードウェアを自動的に合成する**高位合成 (high-level synthesis)** が注目されている。高位合成を行う処理系は、C言語のようなプログラミング言語から VHDL のようなハードウェア記述言語に変換するコンパイラの体裁をとるものが多いが、従来、高位合成の研究は、コンパイラにけるコード最適化や並列化と、VHDL からの合成上の改善とに分けて進められることが多かった。

一方、高位合成を前提にした多くのコード最適化および並列化手法を組み込んだ高位合成器 **SPARK** が、University of California, Irvine (UCI) の S. Gupta らによって提案された。SPARK は、**投機的命令スケジューリング**を中心に、分岐後の命令数のバランスをとってクリティカル経路を短くする**動的分岐バランシング**、容易にループスケジューリングを実現する**ループシフティング**、および投機的スケジューリングによって生じる命令の冗長性を除去する**動的共通部分式除去**といった新しいコード最適化法が、高位合成において有効であることを示した。

当該研究代表者は、カリフォルニア大学アーバイン校で在外研究中に、前述の SPARK が行うコード最適化の多くを同時に実現できる手法を提案したが、既存の手法に特化されている SPARK 上に実現するのは困難であった。

ちょうど同じとき、科学研究調整費によるコンパイラ共通基盤 **COINS** の開発が行われていた。コンパイラ共通基盤は、拡張性を向上させたコンパイラであり、拡張したい部分をモジュールとして付加することによって、短期間で新しいコンパイラを開発し、完全な

コンパイラとして評価をすることができる。コンパイラ共通基盤としては、Stanford 大学のコンパイラグループによって開発された SUIF が有名である。当該研究代表者も、COINS の開発に携わっており、その過程で、高位合成においても共通基盤の必要性を実感するに至った。

## 2. 研究の目的

本研究では、**高位合成向けのコンパイラ共通基盤**の実現を目指して、次の3項目を行う。

- (1) COINS のコード生成部を拡張することによる高位合成器の実現
- (2) 新しい高位合成用のコード最適化手法の研究と、本共通基盤上での実現
- (3) COINS のループ並列化機能とタスク並列化機能を用いた、並列化合成器の実現

## 3. 研究の方法

研究は、次の手順で行う。

- (1) 高位合成における従来法を調査し問題を整理する
- (2) コンパイラ共通基盤 COINS のマシン記述を利用して有限状態マシン制御モデルに基づく VHDL コードを生成できるようにする
- (3) 高位合成向けコード最適化について、既存のコード最適化を実装したのち、新しいコード最適化を開発する。
- (4) COINS が備えている、ループ並列化フェーズとタスク並列化フェーズから得られる結果を反映できるように、これまでに実現した VHDL 向けコード生成部を拡張する

## 4. 研究成果

- (1) COINS コンパイラからの VHDL コード生成 COINS の**目的マシン定義** (以降 TMD) ファイルを変更することによって、関数ごとに、入力プログラムに対する VHDL コードを生成することに成功した。Sparc 用の TMD ファイルを基に、必要な部分を変更することによって、1 カ月程度で実現した。関数呼出しには対応していないが、COINS のインライン展開機能を用いることによって、再帰関数を除く、ほとんど

の関数に対応できる。

(2) 多機能スケジューラの実現

高位合成において、目的ハードウェアの実行効率を向上させるためには、高度に並列化された命令実行が重要である。特に、命令レベルの並列実行を実現する空き資源への**命令スケジューリング**が有効である。高位合成向けの命令スケジューリングとしては、投機的な実行を許して命令をスケジュールする投機的スケジューリング、分岐の制御構造において、命令数のバランスを取ることによって実行経路を短縮させる動的分岐バランシング、ループ内に依存をもつ命令の繰返しを越えてスケジュールするループシフティングがある。

本研究では、前述のスケジューリングの効果を1つのアルゴリズムで実現する方法を実現した。本手法では、空き資源への命令配置と、**部分冗長除去**という命令を移動させることによって命令の冗長性を除去する方法を交互に適用する。命令の配置を投機的に行えば、投機的スケジューリングになり、投機的スケジューリングによって生じた冗長性は、部分冗長除去本来の機能によって除去される

(動的共通部分式除去)。また、部分冗長除去のプログラムの意味を変更しない移動は、分岐において、バランスをとるのに役立つ。ループ内の空き資源に命令を配置して、部分冗長除去を適用すると、ループシフティングが実現できる。いずれのスケジューリング効果についても、命令を配置したのち部分冗長除去を適用するだけで、移動元の命令が除去可能になるかどうかによって、プログラムの制御構造に関わらず、その効果の有無を判定することができる。

本手法 (coins+new) を COINS 上で実現することによって、評価を行った。その結果、従来の命令スケジューリング法 (coins O2+sched) に比べ、数パーセント実行効率が向上することを確認した (図1, 図2)。

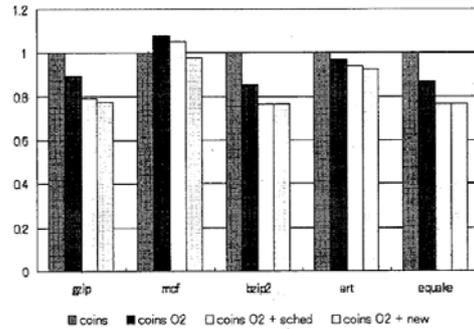
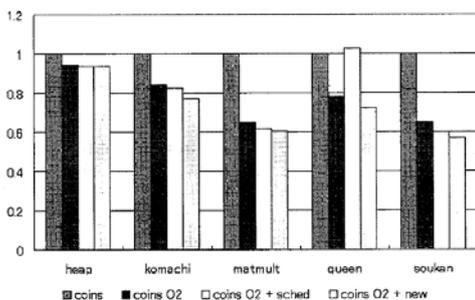


図2. スケジューリング効果 (大きいプログラム)

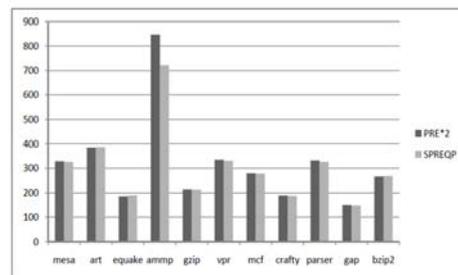
(3) 投機的要求駆動型部分冗長除去の実現

部分冗長除去は、プログラムに対して網羅的に適用する最適化なので、各命令ごとに適用する命令スケジューリングとして用いるには、コストが高い。そこで、命令ごとに必要に応じて適用できる**要求駆動型部分冗長除去法**を開発した。

本手法は、従来の単方向データフロー解析とクエリを伝播させて解析を行う**要求駆動型データフロー解析**を組み合わせることによって実現した。また、本手法は、各命令ごとに適用するので、プログラム中にループが存在すると、クエリがクエリの生成元に戻ってくるという性質がある。このことを用いて、ループ構造が存在する部分にだけ、投機的な移動を許すことができる。従来、部分冗長除去では、ループ構造だけを特別扱いすることはできなかった。

本手法 (PREQP) を COINS 上に実現し、制御フローグラフの開始節に近い命令から順に適用して評価を行った結果、従来法 (PRE\*2) と同等のコストで、プログラムの実行効率を 17%以上向上させることができる場合があることを確認した (図3)。

(4)



る。投機的スケジューリングが部分冗長除去によって実現できたように、逆向きスケジューリングは、**部分無用コード除去法**によって、実現することができる。しかしながら、部分無用コード除去は、プログラム全体に対して網羅的な解析が必要であり、さらに効果を高めるには繰返し適用が必要なので、コストが高いことが知られている。

本研究では、各命令ごとに必要に応じて適用できる部分無用コード除去を実現した。本手法は、制御フローグラフの終了節に近い命令から順に適用することによって、従来、解析の繰返しによって得られた効果を得ることができる。

本手法 (DDPDE) を COINS 上に実現し、SPEC ベンチマークによって評価を行った。その結果、従来法 (DCE) に比べ、プログラム実行効率を 4 パーセント程度向上させることができる場合があることを確認した (図 4)。

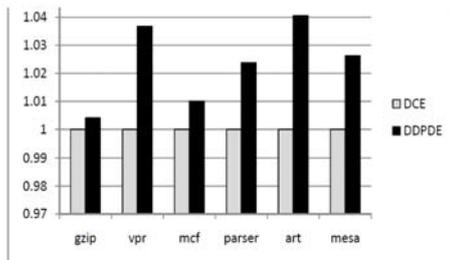


図 4. DDPDE の効果

- (5) アフィン変換によるループの Do-all 化  
 本研究では、COINS が備えている Do-all ループへの並列化機能を強化するために、ループ内の配列アクセス形式を**アフィン変換**によって変形する手法を開発した。  
 本手法によって、従来、データ依存が存在して、並列化できなかったループを依存のない形に変形し、並列化を適用できる機会を増加させることに成功した。

#### 5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

[雑誌論文] (計 5 件)

- ① 滝本宗宏：質問伝播に基づく投機的部分冗長除去，情報処理学会論文誌 プログラミング，Vol. 2, No. 5，pp. 15-27 (Nov. 2009) . 査読有り
- ② 滝本宗宏：要求駆動型部分無用コード除去，組込みシステムシンポジウム2008，

pp.107-114 (Oct. 2008). 査読有り

- ③ 滝本宗宏，佐々政孝：静的単一代入形式を用いた最適化 (発展編)，コンピュータソフトウェア，Vol. 25, No. 1, pp. 30-46 (Jan.2008). (解説論文) 査読有り
- ④ 佐々政孝，滝本宗宏：静的単一代入形式を用いた最適化 (導入編)，コンピュータソフトウェア，Vol. 25, No. 1, pp. 19-29 (Jan.2008). (解説論文) 査読有り
- ⑤ 勝原達也，滝本宗宏：部分冗長除去に基づく大域命令スケジューリング，情報処理学会論文誌 . プログラミング，Vol. 48, No. 12, pp. 52-65 (Aug.2007) . 査読あり

[学会発表] (計 2 件)

- ① 太田真敬，滝本宗宏：COINSを用いた大域的データサイズ推論の実現，日本ソフトウェア科学会大会論文集，第 25 回，4A-2 (2008年9月11日)，筑波大学東京キャンパス。
- ② 滝本宗宏，佐々政孝：質問伝播に基づく要求駆動型大域値番号付け，日本ソフトウェア科学会第10回プログラミングおよびプログラミング言語ワークショップ (PPL2008) 論文集，pp. 233-245 (2008年3月7日)，仙台市。

[図書] (計 1 件)

- ① 中田育男，渡辺坦，佐々政孝，滝本宗宏：コンパイラの基盤技術と実践 - コンパイラ・インフラストラクチャCOINS を用いて，朝倉書店，260 ページ (Jun. 2008)。

#### 6. 研究組織

(1) 研究代表者

滝本 宗宏 ( TAKIMOTO MUNEHIRO )

東京理科大学・理工学部・講師

研究者番号：00318205