

機関番号：34504

研究種目：基盤研究（C）

研究期間：2008～2010

課題番号：20500058

研究課題名（和文）ソフトウェア互換な専用ハードウェアを合成する高位合成系に関する研究

研究課題名（英文）High-Level Synthesis System Autogenerating Software Compatible Hardware

研究代表者

石浦 菜岐佐 (ISHIURA NAGISA)

関西学院大学・理工学部・教授

研究者番号：60193265

研究成果の概要（和文）：

本研究では、ANSI-C で書かれたプログラムに対し、その中の指定された部分を、CPU でソフトウェアとして実行される他の部分からアクセラレータとして起動できるハードウェアに合成する方法について研究を行った。高位合成システムを試作し、FPGA ボード上で動作を確認できた。また、サイクル数がオペランドやハードウェアの状態に依存して変動する演算の存在下で、無駄な待ちを生じさせない「可変スケジューリング」とその近似計算法を開発した。

研究成果の概要（英文）：

We have developed new methods of synthesizing hardware, from specified parts of ANSI-C programs, which can be called from the other parts of the programs as accelerators. We implemented a prototype of a high-level synthesizer and verified the behavior of the generated hardware on FPGA boards. We have also developed a novel concept of *variable scheduling*, which enables efficient adaptive execution under the existence of operations whose latencies vary depending on the values of operands or hardware status.

交付決定額

(金額単位：円)

	直接経費	間接経費	合計
2008 年度	1,300,000	390,000	1,690,000
2009 年度	1,100,000	330,000	1,430,000
2010 年度	1,200,000	360,000	1,560,000
年度			
年度			
総計	3,600,000	1,080,000	4,680,000

研究分野：総合領域

科研費の分科・細目：情報学 ・ 計算機システム・ネットワーク

キーワード：組み込みシステム、高位合成、動作合成、

1. 研究開始当初の背景

大規模化・複雑化するVLSIの設計効率化のため、あるいは、ソフトウェアとハードウェアからなるシステムの設計の効率化のため、

め、プログラミング言語からハードウェア（論理合成可能なレジスタ転送レベル回路）を生成する「高位合成技術」の実用化が模索されている。

C言語（を制限・拡張したもの）からハー

ドウェア単体を合成する技術自体は成熟して来たと言え、eXCite等の商用ツールも存在する。しかし、実際のシステムの設計に適用するには、ポインタや動的データの受け渡しが合成の障害となって記述の大幅修正を余儀なくされたり、ハードウェアとソフトウェアのインタフェースを個々に設計しなければならない等の問題が生じる。このような問題を解決すべく、次のような研究が行われている。

- ・Stanford大学のSpCはポインタを用いた変数のアクセスおよび動的記憶割当の高位合成手法を実現している。しかし、この技術も単体のハードウェア合成に限定されるため、ポインタを用いてソフトウェアやハードウェア間でデータを受け渡すことはできない。

- ・特開2003-114914では、Cプログラムの一部をハード化してソフトウェアから起動する方法として、CPUとハードウェア間に置いた専用のバッファを介してデータ授受を行う方法を提案している。しかしデータ授受は「値渡し」に限られ、またハードウェアが別のハードウェアやソフトウェアを起動することはできない。

- ・特開2003-208311では、ハードウェアをC++のオブジェクトとして扱う「HwObject」の概念を提案している。ソフトウェアがハードウェアにポインタでデータを渡すことが可能な枠組みだが、ハードウェアの合成技術そのものではなく、またハードウェアが別のハードウェアやソフトウェアを起動したり、ハードウェアが動的オブジェクトを割当ててすることはできない。

このように、既存の高位合成技術では、ソフトウェアとして開発されたCプログラムをそのままハードウェアに合成し、従来のソフトウェアと同じように呼び出して利用するということができない。即ち、既存の高位合成技術は、ハードウェア設計言語の延長としてのC言語を対象とするものであり、また、ソフトウェアとハードウェアのインタフェースは別途設計されるものという位置づけである。

このような状況において、我々は、Cプログラム中の指定された関数をそのままハードウェアに合成し、これがソフトウェアとしてCPUで実行される関数や、ハードウェア化される別の関数との間で相互呼び出し可能となるような、高位合成システムの枠組みを提唱している。その特長は次のとおりである。

- (1) ソフトウェアとハードウェアが論理アドレス空間を共有する。これにより、ポインタや動的オブジェクトへのアクセスを含むCの関数を自然な形でハードウェアに合成できる。論理アドレス空間の共有は

「アクセス調停回路」により行う。この回路は、CPUとハードウェアからのメモリアクセス要求を調停するとともに、データのキャッシュ、およびCPUとハードウェア間の制御の効率的な受け渡しも行う。

- (2) 関数呼び出しの枠組みを維持したままソフトウェアとハードウェアの相互起動が可能である。特に、ハードウェア化された関数からソフトウェア関数を起動できるため、ハードウェアが動的領域の確保や入出力を行える。

この枠組みにより、仕様やリファレンスとして与えられるCプログラムの大幅改変なしに、ソフトウェアとハードウェアからなるシステムが設計でき、高位合成を適用可能な範囲を大幅に拡大することができる。期待できる。

2. 研究の目的

本研究は、VLSIの「高位合成」の考え方を更に進め、ANSI-Cプログラムを入力として、それを実行する汎用プロセッサと等価な専用ハードウェアを自動生成する「ソフトウェア互換ハードウェア合成」を実現するための基盤技術の研究を行う。本手法の特長は、1) ハードウェア設計用に制限・拡張されたC言語ではなく、ANSI-Cで書かれたプログラムをほぼそのままハードウェア化できること、2) 合成された専用ハードウェアが他のソフトウェア部品相互に関数呼び出しの形で起動し合うことができること、および、3) ポインタを用いて動的オブジェクトの共有やデータ参照渡しができることにある。これによって、ハードウェア設計自動化の適用範囲を、従来の高位合成から飛躍的に拡大することを狙いとしている。

本研究では、このような新しい高位合成の枠組みを確立するための基板研究を行うことを目的としている。具体的には、ソフトウェアとハードウェアのインタフェース、C言語の種々の記憶クラスの変数へのアクセスをハードウェアとして実現する方法、ソフトウェアとハードウェアの実行の並列化、およびこれらの研究に基づくプロトタイプシステムの実装とその評価に関する研究を行う。

3. 研究の方法

(1) 具体的な研究項目

下記4項目に分けて研究を行った。

① ANSI-C からのハードウェア合成手法

「ソフトウェア互換ハードウェア」を実現するための基盤になる技術として、ANSI-C で書かれたプログラムをできる限りそのままハードウェアに合成する手法の研究を行った。

基本的には、ハードウェアもメモリにアクセスし、CPU 上で動作するプログラムと同じ動作をするようにハードウェアを合成するという考えに基づけばよい。処理系の具体的な構築法としては、次の2つについて検討およびプロトタイプを試作を行った。

1) SUIF を用いる方法

SUIF は Stanford 大学で開発された並列化コンパイラの中間表現である。C プログラムは解析されて SUIF フォーマットと呼ばれる中間表現に変換されるので、これを入力として高位合成用の中間表現 CDFG (control dataflow graph) を生成すれば、高位合成系を構築することができる。

2) アセンブリを入力とする方法

既存のCコンパイラが生成するアセンブリコードを入力とし、ここから CDFG を構築する。結合する CPU と同じコンパイラを用いることにより、内部データ構造の齟齬を回避できるという利点がある。

C 言語を CDFG に変換するフロントエンドは上記の2つを実装するが、CDFGを入力として Verilog-HDL を生成するバックエンドの処理は共通に利用できるようにした。

② ソフトウェアとハードウェアの連携法

合成したハードウェアをソフトウェアから起動する方法に関しては、グローバル変数を用いたポーリング方式を提案しているが、さらに汎用性が高く効率の良い方法について検討を行った。

③ 高位合成アルゴリズム

スケジューリング、バインディング等の高位合成のバックエンドのアルゴリズムの研究を行った。特に、メモリアクセスや逐次方式の乗除算のように、実行サイクル数がオペランドやその時点でのハードウェアの状況によって変動する演算が存在する場合に、効率よく実行が行えるスケジューリングについて検討を行った。

④ システムの試作とハードウェアの検証

上記の方式に基づく高位合成システムのプロトタイプの開発を進めた。これと並行し、CPU と合成されたハードウェアを結合したものの動作を、RTL シミュレーションおよび FPGA のボード上で検証した。

(2) 研究組織と役割分担

石浦 (研究代表者) は、研究の統括を行うとともに、同所属の研究協力者とともに合成システムの方式、アルゴリズム、システム試作の研究を行った。

神原 (研究分担者) は、合成されるハードウェアとプロセッサの接続に関する研究とハードウェアレベルでの評価を行った。

富山 (研究分担者) はソフトウェアとハードウェアの連携方式の研究とシステム全体の評価を行った。

3名が上記の役割を分担し、1ヵ月に1回の頻度で進捗報告と全体の検討を行うという形態で研究を推進した。

4. 研究成果

(1) ANSI-C からのハードウェア合成法

ANSI-C で書かれたプログラムをハードウェアに合成する手法として、SUIF を用いる方法と、既存のCコンパイラが生成するアセンブリコードを入力とする方法の2通りについて、高位合成システムの試作を行った。

SUIF の中間表現を経由する方法では、Cプログラムの持つ構造や記憶クラスの情報を利用してハードウェアの最適化を行うことができる。特に、局所変数 (自動変数) はレジスタに、局所配列 (自動変数の配列) はレジスタファイルに、また、定数の配列は ROM に合成することにより、単純な合成法と比較してハードウェアの実行速度も回路規模も大幅に改善できる。

一方で、SUIF を用いる方法では合成できるCプログラムのクラスを ANSI-C に準拠したものにまで拡張することが困難であることも判明した。具体的には、下記のような点が挙げられる。

- ・レジスタに合成する局所変数がポインタにより参照されている場合に正確な合成を行うことが難しい。
- ・合成のための中間表現である CDFG (control dataflow graph) を C 言語の全ての制御構造に対応させることが難しい。
- ・C 言語の全てのデータ型の内部表現を、結合する CPU 用のコンパイラのそれと一致させることが難しい。特に、構造体のパディングや最適化による記憶割当の変更に従うことは極めて難しい。
- ・SUIF 自体が開発から長年を経過しており、標準的な規格の ANSI-C や C99 で書かれたプログラムを処理できないことがある。

このような状況に鑑み、本研究ではもう一つの合成手法として、既存のCコンパイラ、具体的には、合成したハードウェアと結合する CPU である MIPS のコード生成に用い

るのと同じ GCC が生成するアセンブリコードを高位合成の入力として用いる手法について研究を行った。

この方法では、上記に列挙した SUIF の問題をすべて解決することができる。GCC は開発が継続的に行われており、新しい規格への追従性も高く、また産業界で標準的に利用されていて信頼性も高い。更に、コンパイラ自体が非常に多くの最適化処理を行うため、合成系側で全ての最適化処理を実装する必要がないことも大きな利点として挙げられる。

具体的な処理系としては、MIPS との結合を実現するため、GCC によって生成されるリンク済みコードを入力とし、これを逆アセンブルして得られるアセンブリコードから合成対象となる関数を抽出し、それを CFG に変換して合成処理を行う方式でシステムを試作した。

この方法により、SUIF を用いる場合と比較して非常に広範な C プログラムを合成できるようになった。しかし一方で、入力となるアセンブリプログラムからは C 言語レベルの情報が失われているため、ハードウェアの最適化に限界が生じるというデメリットは存在する。具体的には、変数が局所配列や定数配列であることの利用や、ループ構造の認識ができないことである。上位構造の推定やプラグマ記述等によってこの点を改善していくことが今後の課題として挙げられる。

(2) ソフトウェアとハードウェアの連携法
合成したハードウェアを CPU 上で実行されるソフトウェアから起動する方法に関しては、ハードウェアを関数として呼び出す方法と、ハードウェアをコプロセッサとして起動する方法の2通りを開発した。

① 関数としてのハードウェアの起動

従来より提案している通り、ハードウェア化された関数のソフトウェアからの起動は、引数と戻り値の授受と起動・終了待ちの制御をグローバル変数を介して行うことにより実現できる。本研究では、これに基づき更に本手法の適用性や結果得られるシステムの効率を高める次の2つの方法を開発した。

1) ソフトウェアからのハードウェア呼び出しだけでなく、ハードウェアからソフトウェアの関数を呼び出す方法を開発した。

2) プラグマを用いることより、個々の関数をハードウェアとソフトウェアのどちらで実装するかの指定、非同期関数呼び出しによるハードウェアとソフトウェアの並列実行を可能にする枠組みを構築した。

② CPUと密結合したコプロセッサ

従来提案されている方法とは全く異なり、ほぼオーバヘッド無しにソフトウェアからハードウェアに制御を切り替える手法を開発した。

この手法では、関数単位ではなく、CPU で実行される機械語の任意区間を切り出してハードウェア化する。合成されたハードウェアは、主記憶だけでなく、レジスタファイル、およびそれにアクセスするフォワードイングユニットも CPU と共有する。コプロセッサは CPU の PC (プログラムカウンタ) を監視し、ハードウェア化された機械語の先頭番地に達すると CPU から制御を取る (ハードウェアの実行中は、CPU は NOP を実行し続ける)。通常のコプロセッサの起動とは異なり、ハードウェアは CPU のレジスタファイルに直接アクセスしてデータの授受を行う。ハードウェアは実行が終了すると、PC の値を区間の最終番地に設定し、CPU 制御を戻す。

この手法では、機械語の任意の区間 (ループを含んでも良い) をハードウェア化して高速化できる上、ハードウェアを起動するオーバヘッドがほぼゼロである。さらに、CPU が実行する機械語は一切変更する必要はないため、元々の CPU を CPU +ハードウェアにそのまま差し替えることも可能である。

この方式に基づいて、MIPS とハードウェアを結合する枠組みを構築し、RTL シミュレーションおよび FPGA ボード上で動作確認を行った。現在、単一および2つのハードウェアで動作の確認が行えている。

(3) 不定サイクル演算のスケジューリング

実行サイクル数がオペランドやハードウェアの状況に依存して変動する演算が存在する場合に、実行時にスケジューリングを適応的に変化させることにより実行効率を向上させる「可変スケジューリング」の手法を新たに考案した。

これは、従来の高位合成における「制御ステップ」を一つの状態ととらえ、演算器から出力される完了信号に依存して次状態を変更する状態遷移グラフによって計算の過程を表現することにより実現する。

この方法により、不定サイクル演算に対して最大値や最小値を仮定した従来のスケジューリング手法に比べて、実行サイクル数を数%短縮できることが確認できた。しかし一方で、状態遷移グラフのサイズが入力となる DFG (dataflow graph) の増大に伴って急激に増え、合成される回路の制御系とマルチプレクサのための回路規模や遅延が著しく増大する場合が生じることが判明した。

この問題を解決するため、実行サイクル数を犠牲にして回路規模を抑制する、可変

スケジューリングの近似手法を考案した。手法は2通りあり、1つは一定以上の実行サイクル数削減に貢献しない状態を生成しない方法、もう1つは1演算を複数の演算器にバインディングしなければならない状態を生成しない方法である。

この2手法に基づくスケジューラを実装し、FPGAを対象にした論理合成まで行って性能評価を行った結果、近似手法が回路規模やレジスタ間遅延の抑制に有効であることを確認できた。

(4) 合成システムの試作と動作の検証

高位合成系のプロトタイプの開発を進めるとともに、CPUと合成されたハードウェアを結合して動作確認を行う環境を作成した。

高位合成システムは、GCC (mips-elf-gcc 4.5.1) と Perl5 で実装し、Ubuntu Linux, Cygwin, MacOSX 上で動作する。

MIPS R2000 互換 CPU に対して合成したハードウェアを3個まで結合できるインタフェースを実装し、RTL シミュレーションおよびFPGA上での動作確認を行った。

作成したFPGAボードは、MIPSプロセッサの動作を理解するための教材としても活用している。

5. 主な発表論文等

[雑誌論文] (計2件)

① Y. Toda, N. Ishiura, and K. Sone: "Static Scheduling of Dynamic Execution for High-Level Synthesis," *Proc. the Workshop on Synthesis and System Integration of Mixed Information Technologies*, 査読有り, pp. 107-112 (Mar. 2009).

② M. Nishimura, N. Ishiura, Y. Ishimori, H. Kanbara, and H. Tomiyama: "High-Level Synthesis of Software Function Calls," *IEICE Trans. Fundamentals*, 査読有り, No. E91-A, pp. 3556-3558 (Dec. 2008).

[学会発表] (計7件)

① 曾根康介, 石浦菜岐佐: 高位合成における可変スケジューリングの近似手法, 電子情報通信学会 VLSI 設計技術研究会, 慶応義塾大学 (神奈川県), 2011年1月17日.

② 入谷賢孝, 池上達也, 石浦菜岐佐, 神原弘之, 富山宏之: MIPS アセンブリを中間表現とする高位合成システムの実装, 情報処理学会組込みシステム研究会, 八丈シーパークリゾート (東京都), 2010年3月28日.

③ 戸田勇希, 石浦菜岐佐, 神原弘之, 富山

宏之: CPU と密に結合したコプロセッサに基づくハードウェア/ソフトウェア協調設計, 情報処理学会組込みシステム研究会, 株式会社東芝 (東京都), 2010年1月29日.

④ 曾根康介, 石浦菜岐佐, 入谷賢孝, 戸田勇希: 高位合成における可変スケジューリングの近似アルゴリズム, 情報処理学会関西支部大会, A-06, 神戸大学 (兵庫県), 2009年9月29日.

⑤ 神原弘之, 金城良太, 戸田勇希, 矢野正治, 小柳滋: パイプラインプロセッサを理解するための教材 RUE-CHIP, 情報処理学会関西支部大会, A-09, 神戸大学 (兵庫県), 2009年9月29日.

⑥ 石守祥之, 石浦菜岐佐, 富山宏之, 神原弘之: 高位合成システム CCAP の AMP マルチコアシステム設計のための拡張, 電子情報通信学会 VLSI 設計技術研究会, 慶応義塾大学 (神奈川県), 2009年1月29日.

⑦ 戸田勇希, 石浦菜岐佐, 曾根康介: 不定サイクル演算を考慮した高位合成の可変スケジューリング・バインディング, 電子情報通信学会 VLSI 設計技術研究会, 北九州市学術研究都市 (福岡県), 2008年11月19日.

6. 研究組織

(1) 研究代表者

石浦 菜岐佐 (ISHIURA NAGISA)
関西学院大学・理工学部・教授
研究者番号: 60193265

(2) 研究分担者

神原 弘之 (KANBARA HIROYUKI)
(財)京都高度技術研究所・研究開発部・主任研究員
研究者番号: 80373497

富山 宏之 (TOMIYAMA HIROYUKI)
立命館大学・理工学部・教授
研究者番号: 80362292