テクニカルノート

充足可能性判定を利用した最適コード生成手法

瀬戸謙修[†] 藤田昌宏[†] 浅田邦博[††]

充足可能性判定（SAT）を利用した最適なコード生成手法を提案する．まずコード生成問題を有限状態機械（FSM）を利用して定式化する（瀬戸ほか，2002）．次にFSMを組合せ回路へと展開し（Biereほか，1999），最適コード生成問題を充足可能性判定（SAT）問題として解く．最新のSATソルバ（Moskewiczほか，2001）を使用した実験結果から，SATベースのコード生成手法がBDDベースのコード生成（瀬戸ほか，2002）よりも効果的であることが示される．

Optimal Code Generation Based on Boolean Satisfiability

Kenshu Seto,[†] Masahiro Fujita[†] and Kunihiro Asada[††]

We present a method for optimal code generation based on Boolean satisfiability (SAT). First, code generation is formulated based on a finite state machine (FSM) (Seto, et al., 2002). Next, we unroll the FSM to a combinational circuit (Biere, et al., 1999), so that optimal code generation problem is solved as a SAT problem. Experimental results using a state-of-the-art SAT solver (Moskewicz, et al., 2001) demonstrate that SAT-based optimal code generation is more effective than BDD-based symbolic approach (Seto, et al., 2002).

1. はじめに

システムLSIでは，要求性能を満足させると同時に設計の柔軟性を高めるため，プロセッサを含んでいることが多い．しかし従来のコード生成では，コード選択，レジスタ割当て，スケジューリング等を独立に実行しており，専用レジスタを持つ命令レベル並列アーキテクチャに対して最適コードを生成できない．この問題に対処するため，有限状態機械（FSM）およびその二分決定木（BDD）を使用したシンボリックな解析[6]により，最適コードを生成する手法が提案された[1]．しかし非常に小規模なプログラムや比較的簡単な命令セットしか扱えていない．近年，FSMの限定モデル検査に充足可能性判定（SAT）を利用した方が，BDDを利用するのに比べて有利な場合があることが報告された[2]．それを受けて本稿では，コード生成用FSM[1]を修正し，SATベースの限定モデル検査を適用することで，ステップ数最適コード生成問題をSAT問題として解く．実験結果を示し，SATベースのコード生成の有効性を示す．

なお本稿と似たアプローチとして，整数線形計画法を利用したコード生成手法[4),5)]がある．

2. コード生成用FSM

FSMによるコード生成の入力ファイルは，データフローグラフ（DFG）および命令パターンである．DFG中のノードは操作を表し，エッジは操作間のデータ依存性を表す（**図1**(a)）．命令パターンはプロセッサの命令をグラフ表現したものである（図1(b)）．まずこれらのグラフ間でマッチングを行い，各DFGノードの操作を実行可能な命令の候補（これをマッチと呼ぶ）を列挙する（図1(c)）．

図1(a)中に示した変数 $a_{1,reg} \in \{0,1\}$ は，コード生成用FSM（以下，**CGFSM**と略記する）の状態ビットの1つで，DFGノード1の操作結果がロケーション reg に存在するかどうかを示し，1，0でそれぞれ存在，非存在を表す．図1(c)に示した変数 $e_1 \in \{0,1\}$ は，CGFSMの入力ビットの1つで，各ステップでマッチ1を実行するかどうかを示し，1，0でそれぞれ実行，非実行を表す．CGFSMの状態遷移の一例として，あるステップで入力ビットが $e_1 = 1$

† 東京大学工学部電子工学科
Department of Electronics Engineering, The University of Tokyo
†† 東京大学大規模集積システム設計教育研究センター
VDEC, The University of Tokyo

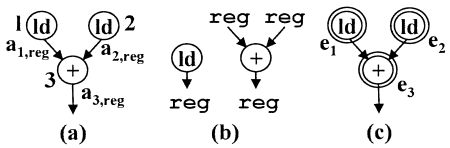


図 1 DFG，命令パターンおよびマッチの例
Fig. 1 Example of DFG, instruction pattern and match.

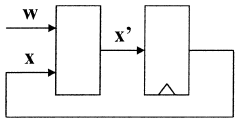


図 2 有限状態機械（FSM）
Fig. 2 Finite State Machine (FSM).

のとき，次のステップで状態ビットが $a_{1,reg} = 1$ と変化する．最終的に図 1 の DFG および命令パターンに対して CGFSM の入力 $\mathbf{w}$ および状態 $\mathbf{x}$ を書き下すと次のようになる．

$$\mathbf{w} = (e_1, e_2, e_3, f_{1,reg}, f_{2,reg}, f_{3,reg})$$
$$\mathbf{x} = (a_{1,reg}, a_{2,reg}, a_{3,reg}, v)$$

$f_{n,l}$ および v はそれぞれ，本稿で新規に CGFSM に導入した入力変数および状態変数である．たとえば $f_{1,reg} \in \{0,1\}$ は，CGFSM の入力ビットの 1 つで，次のステップで状態ビット $a_{1,reg}$ をリセットするかどうかを示し，1，0 でそれぞれリセット，非リセットを表す．ただし $f_{1,reg}$ と e_1 が同時に 1 のときは，e_1 を優先し $a_{1,reg}$ をセットする．$v \in \{0,1\}$ は，リソース制約，データ依存制約，レジスタ制約の制約違反がどれか起こった場合に 1 にセットされる状態ビットで，一度セットされたらリセットされることはない．CGFSM の初期状態 $\mathbf{x_0}$ を $(0,0,0,0)$，最終状態 $\mathbf{x_f}$ を $(-,-,1,0)$ と定める．ここで最終状態中の文字 $'-'$ はドントケアを表す．CGFSM に 3 章で説明するモデル検査を適用することで，初期状態から最終状態までの最短経路を探索し，そのときの入力列 $\mathbf{w_0}, \mathbf{w_1}, \ldots, \mathbf{w_f}$ をもとに，ステップ数最適なコードを抽出可能である．CGFSM の詳細は文献 1) に記載されており，ページ数の都合で本稿では割愛する．

3. BDD を利用した FSM の解析

図 2 に FSM の模式図を示す．図中で $\mathbf{w}$，$\mathbf{x}$，$\mathbf{x'}$ はそれぞれ FSM の入力，現状態，次状態を表す．モデル検査を利用すると，順序回路の状態 $\mathbf{x}$ に関する論理式 $F(\mathbf{x})$ の成立あるいは不成立を検査することができる．不成立の場合，そこに至る入力列 $\mathbf{w_0}, \mathbf{w_1}, \ldots, \mathbf{w_f}$ が出力される．

BDD ベースのモデル検査では，次の式を使用して

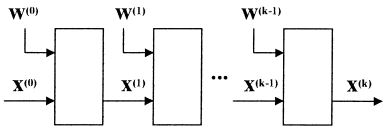


図 3 FSM を展開して生成した組合せ回路
Fig. 3 Combinational circuit expanded from FSM.

シンボリックに FSM の状態探索を行う[6]．

$$S_{i+1}(\mathbf{x}') = \exists \mathbf{x}, \mathbf{w}.(T(\mathbf{x}, \mathbf{w}, \mathbf{x}') \cdot S_i(\mathbf{x})) \quad (1)$$

$S_{i+1}(\mathbf{x}')$，$S_i(\mathbf{x})$，$T(\mathbf{x}, \mathbf{w}, \mathbf{x}')$ はそれぞれ次状態の集合，現状態の集合，状態遷移関係を表す論理関数で，BDD で表現される．各 BDD 表現がコンパクトな場合，式 (1) によって次状態の集合を次々に求めていくことにより，効率的に最終状態を探索できる．文献 1) では，CGFSM に対し BDD ベースのモデル検査を適用したが，比較的簡単な命令セットの場合でも DFG ノード数が 20 程度までのプログラムが限界であった．

4. SAT を利用した FSM の解析

文献 2) によって，限定モデル検査を SAT 問題として解くアプローチの有効性が示された．限定モデル検査とは，最大 k ステップまでに到達可能な状態のみを検査対象とするモデル検査である．**図 3** は，図 2 の FSM からフリップフロップを取り除き，状態遷移関数を表す組合せ回路を k ステップ分展開することで得られた組合せ回路である．この入力は $(\mathbf{w}^{(0)}, \ldots, \mathbf{w}^{(k-1)}, \mathbf{x}^{(0)})$ で，出力は $\mathbf{x}^{(k)}$ である．$\mathbf{w}^{(i)}$，$\mathbf{x}^{(i)}$ はそれぞれステップ i での FSM への入力および状態を表す．図 3 に示すような組合せ回路に関する SAT 問題を解くことにより，限定モデル検査が可能である．SAT を利用したコード生成では，CGFSM に対して SAT による限定モデル検査を適用し，k ステップの最短入力列を $\mathbf{w}^{(0)}, \ldots, \mathbf{w}^{(k-1)}$ を求め，その入力列から最短ステップのコードを抽出する．ステップ数 k は最初 0 に設定し，1 ずつ増やしていきながら CGFSM を最終状態に到達させるような最短の入力列を探索する．

5. 実験結果

本章では SAT を利用したコード生成の実験結果を示す．限定モデル検査プログラムとして，VIS[7] バージョン 2.0 を利用した．SAT ソルバとして Chaff[3] を使用した．実験に使用したマシンは，1.1 GHz Celeron（メモリは 2 G バイト）である．実験では**図 4** に示すような，アナログデバイス社の DSP である ADSP-2100 とほぼ同様のデータパスに対するコード生成を行った．並列命令および専用レジスタを持つアーキテクチャで，文献 1) におけるターゲットアーキテクチャよりも専

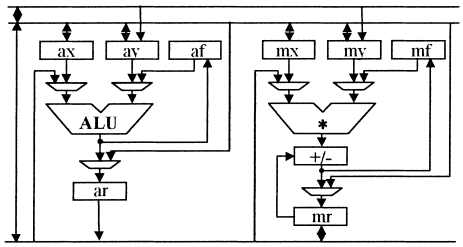


図 4 実験に使用したアーキテクチャ
Fig. 4 Architecture used for experiment.

表 1 コード生成結果
Table 1 Code generation result.

program	nodes	steps	time(s)
cm	10	6	36
iir	17	8	135
cu	13	7	105

表 2 SAT 問題の規模
Table 2 Size of SAT instances.

program	variables	clauses
cm	204190	594927
iir	500546	1464453
cu	362107	1059144

用レジスタの種類が多く複雑である．

表 1 に SAT ベースのコード生成手法によるコード生成結果をまとめる．表中の各列は左から順にそれぞれ，プログラム名，DFG ノードの数，最適アセンブリコードのステップ数，コード生成に要した CPU 時間（秒）を表す．なお BDD ベースのコード生成では，いずれの例に対してもメモリを 1 G バイト以上消費しても，最適コードを生成することができなかった．

表 2 にコード生成を行うために解いた SAT 問題の規模をまとめる．表中の各列は左から順にそれぞれ，プログラム名，変数の数，節の数を表す．SAT 問題の規模はステップ数 k を増やしていくにつれて，ほぼ比例して増大していくため，表 2 では SAT 問題の最大規模を示してある．なお SAT 問題の規模の大部分（およそ 80%以上）は，リソース制約等の制約条件を表現するための閾値関数の SAT 表現に由来するものであり，この部分の効率的な取扱いは今後の課題である．

6. 結　　論

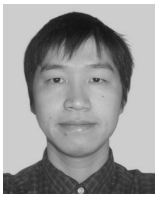
本稿では，コード生成用 FSM に SAT ベースの限定モデル検査を適用することにより，SAT ベースのコード生成を行った．実験結果から，BDD ベースのコード生成では最適コードを生成できない例でも，SAT ベースのコード生成では最適コードを生成することを示し，SAT ベースのコード生成の有効性を示した．

今後の課題として，ASAP（As Soon As Possible），ALAP（As Late As Possible）等のスケジューリングバウンドを利用して不要な変数を削減し，最適コード生成の時間をさらに短縮することがあげられる．また必ずしも最適でなくても，最適に近い解をより短期間に与えるようなヒューリスティックを考案することも重要と考えている．文献 8) で提案された方法でリソース制約条件を扱うことで，SAT 問題の規模が小さくなり，本手法が適用範囲が広がると考えている．また本稿の手法により，SIMD 命令を持つプロセッサのコード生成を行うことも考えている．

参 考 文 献

1) 瀬戸謙修，藤田昌宏：有限状態機械（FSM）とシンボリック状態探索を利用したコード生成手法，情報処理学会論文誌，Vol.43, No.5, pp.1235–1251 (2002).
2) Biere, A., Cimatti, A., Clarke, E.M., Fujita, M. and Zhu, Y: Symbolic Model Checking using SAT procedures instead of BDDs, *Proc.36th Design Automation Conf.*, pp.317–320 (1999).
3) Moskewicz, M., Madigan, C., Zhao, Y., Zhang, L. and Malik, S.: Chaff: Engineering an Efficient SAT Solver, *Proc. 38th Design Automation Conf.*, pp.530–535 (2001).
4) Wilson, T., et al.: An ILP-Based Approach to Code Generation, *Code Generation for Embedded Processors*, pp.103–118, Kluwer Academic Publishers (1995).
5) Kästner Daniel: PROPAN: A Retargetable System for Postpass Optimisations and Analyses, *Proc. ACM SIGPLAN Workshop on Languages, Compilers and Tools for Embedded Systems* (*LCTES 2000*), pp.63–80 (2000).
6) Burch, J.R., Clarke, E.M., McMillan, K.L., Dill, D.L and Hwang, L.J.: Symbolic Model Checking: 10^{20} states and beyond, *Information and Computation*, Vol.98, No.2, pp.142–170 (1992).
7) Brayton, R.K., et al.: VIS: A system for verification and synthesis, *Proc. 8th Conference on Computer Aided Verification* (*CAV'96*), pp.428–432, LNCS 1102, Springer-Verlag (1996).
8) Aloul, F.A., Ramani, A., Markov, I. and Sakallah, K.: Generic ILP versus Specialized 0-1 ILP: an Update, *Proc. Intl. Conf. on Computer-Aided Design 2002* (*ICCAD 2002*) (2002).

(平成 14 年 10 月 23 日受付)
(平成 15 年 1 月 7 日採録)



瀬戸 謙修

1997年東京大学工学部電気工学科卒業．1997年～1998年カリフォルニア大学バークレイ校CADグループ交換留学．1999年東京大学大学院工学系研究科電子工学専攻修士課程修了．2000年～2002年パシフィック・デザイン株式会社勤務．C言語からのシステムLSI設計環境開発およびRISCプロセッサのコンパイラ開発に従事．現在，東京大学工学系研究科電子工学専攻博士課程在学中．主に論理合成，アプリケーション特化型プロセッサ向けコンパイラの研究に従事．

藤田 昌宏（正会員）

1985年東京大学大学院工学系研究科情報工学博士課程修了．工学博士．同年富士通入社．富士通研究所にて，VLSI CADの研究に従事．1988年～1989年イリノイ大学客員研究員．1993年米国富士通研究所出向．VLSI CAD研究グループの立ち上げ．2000年より東京大学大学院工学系研究科電子工学専攻教授．論理合成，論理検証，システムレベル設計支援技術等の研究に従事．SpecCコンソーシアム言語ワーキンググループ主査．IEEE，ACM会員．

浅田 邦博

1952年福井市生まれ．1980年3月東京大学工学系研究科電子工学博士課程修了（工学博士）．1980年4月より東京大学に任官．同講師，同助教授をへて1995年東京大学工学系研究科教授（電子工学科）．1996年東京大学大規模集積システム設計教育研究センター（VDEC）の設立にともない異動，現在に至る．1985年～1986年英国エディンバラ大学訪問研究員．1990年～1992年電子情報通信学会英文誌エレクトロニクス（初代）エディタ．2001年～2002年IEEE SSCS Japan Chapter Chair．専門，集積システム・デバイス工学．著書,「アナログ電子回路」（昭光堂，1998),「VLSIの設計I，II」（共著，岩波書店，1985）ほか．