

部分グラフ同型判定のためのデータ依存回路の実装と評価

山本昌治[†] 市川周一[†] 山本浩司[†]

部分グラフ同型判定は幅広い応用を持つが, 一般に NP 完全である. Ullmann のアルゴリズムは論理回路で実現できるが, 回路規模が大きく実装困難である. 小西のアルゴリズムは, Ullmann の回路より小さい論理で実現できるが, 実行時間が長い. 一般に論理回路の入力が決まれば論理を単純化して回路規模を縮小できる (データ依存回路). 市川らは, Ullmann の回路のデータ依存回路を検討し, 論理規模が縮小することを報告したが, 実装と評価は行っていない. 本研究では, Ullmann のデータ依存回路 Ud と小西のデータ依存回路 Kd を FPGA で実装評価する. 入力グラフの頂点数が 16 のとき, Ud の性能はソフトウェアの 9.4 倍, Kd は 13.3 倍である. このとき Ud の論理規模は, Kd の 2.2 倍であった.

The Implementation and Evaluation of Data Dependent Hardware for Subgraph Isomorphism Problems

SHOJI YAMAMOTO,[†] SHUICHI ICHIKAWA[†] and HIROSHI YAMAMOTO[†]

Subgraph isomorphism problems have various important applications, while they are generally NP-complete. Though Ullmann's algorithm can be implemented by parallel hardware to accelerate the execution, it requires too many hardware resources. Konishi's algorithm requires much less resources, but it is slower than Ullmann's. Generally, logic gates are reducible if any inputs are fixed to constants. Ichikawa and Yamamoto reported that the data dependent circuits for subgraph isomorphism problems have smaller logic scale than the original, but they did not implement the circuit actually. This study describes the implementations and evaluations of data dependent circuits of Ullmann's algorithm (Ud) and Konishi's algorithm (Kd). For the graphs of 16 vertices, Kd and Ud were 13.3 and 9.4 times faster, respectively. The logic scale of Ud was 2.2 times larger than Kd in this case.

1. 部分グラフ同型判定問題

2つのグラフ G_α と G_β が与えられたとき, G_α が G_β の部分グラフと同型か否かを判定する問題を“部分グラフ同型判定問題”という. 例えば図 1 において, G_α は G_β と同型な部分グラフを持つが, G_γ は持たない. 部分グラフ同型判定問題は幅広い応用を持つが, 一般に NP 完全であり計算困難である.

部分グラフ同型判定には, Ullmann のアルゴリズムが広く用いられている. Ullmann のアルゴリズムは論理回路で実現可能であるが, 回路規模が大きく実装に難がある. 市川ら¹⁾ は, 入力データに依存した論理回路 (データ依存回路) を設計することによって, Ullmann の回路の論理規模が削減できることを示した. ただし, 市川ら¹⁾ は実装評価を行っていない.

本研究の目的は, 部分グラフ同型判定のためのデータ依存回路を, 現実の FPGA(Field Programmable



図 1 部分グラフ同型判定の例

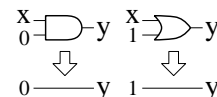


図 2 簡単化の例

Gate Array) 上で実装・評価することである. 本研究では Ullmann のアルゴリズムに加えて, 小西のアルゴリズム²⁾ のデータ依存回路も実装評価の対象とする.

2. データ依存回路

一般に論理回路の入力が定まると, 論理の単純化によりゲートが除去され, 回路規模が小さくなる (図 2). 論理段数の減少により動作周波数の向上も期待でき, またデータ保持のための記憶素子も一部削減できる. このような回路を, 以下“データ依存回路”と呼ぶ.

データ依存回路では, 入力データ毎に回路を生成する必要がある. そこで本研究では FPGA のような再構成可能論理の利用を前提とする. 実際にデータ依存回路を動作させるには, 回路記述生成, 論理合成, テ

[†] 豊橋技術科学大学 知識情報工学系

Department of Knowledge-based Information Engineering, Toyohashi University of Technology

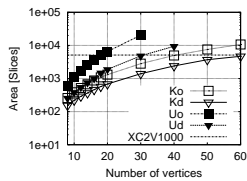


図 3 回路規模

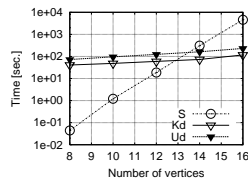


図 4 実行時間

クロジマッピング, 配置配線, 構成データ生成, ダウンロード等の手順が必要である. 結果として, データ依存回路の実行時間は, 回路生成時間と動作時間の合計になる. なお, 論理の簡単化は, 回路記述生成と論理合成の段階で行われる.

3. 設 計

本研究では, 以下の 4 つの場合について, 設計の優劣を比較検討した. 残念ながら紙数の関係上, 各設計の詳細については省略する.

Uo Ullmann の提案した回路 (データ非依存). 市川ら¹⁾ の INDEP と等価な設計.

Ud 入力データ依存の Ullmann 回路. 市川ら¹⁾ の BOTH2 と等価な設計.

Ko 小西のアルゴリズム²⁾ を実装した回路 (データ非依存). ただし枝刈り回路を並列化して実装した点で, 論文 2) の実装と異なっている.

Kd 上記 Ko のデータ依存回路. Ko の入力を定数に置き換えて回路を生成した.

入力には 100 対のグラフ G_α, G_β をランダムに生成して用い, 結果は 100 対の平均値で示す. G_α, G_β の辺密度は, 各 0.3, 0.6 とした. 辺密度 ed とは, グラフの頂点数を p , 辺数を q としたとき, $ed = 2q/p(p-1)$ で定義されるパラメータである ($0 \leq ed \leq 1$).

上記 4 つの設計に関して, 回路記述生成~テクノロジーマッピングまでを行い, 結果を比較した. 回路記述生成には C 言語で実装した自作プログラム, 論理合成には Synopsys FPGA Compiler II, テクノロジマッピングには Xilinx ISE 4.2i を使用し, Athlon XP 1700+ PC 上で動作させた. ターゲットデバイスは Xilinx Virtex-II XC2V1000 FPGA である.

テクノロジーマッピングのログから回路規模 (スライス数) と動作周波数を見積もることができる. さらに C 言語で実装したサイクルレベル・シミュレータで, 同型判定に必要なサイクル数を求めた. こうして得られた動作周波数とサイクル数から, 回路の動作時間を見積もることができる. データ依存回路では, 見積もった動作時間と (マッピングまでの) 回路生成時間の合計を, 実行時間とした. 性能比較の対象として, Ullmann のアルゴリズムを C 言語で実装し, Celeron 1.2GHz PC 上で実行した時間も測定した (以下, ソフトウェア S と表す).

表 1 実装結果 [秒]

	Kd	Ud	S
回路記述生成	0.08	0.18	
論理合成	79.73	181.75	
テクノロジーマッピング	9.51	18.02	
配置配線	82.85	184.15	
構成データ生成	14.60	17.16	
ダウンロード	17.56	17.55	
回路生成時間	204.33	418.81	
動作時間	142.46	72.46	4599.24
実行時間	346.79	491.27	4599.24

平均回路規模を図 3 に示す. Kd の回路規模は Ko の 44%~60% (頂点数 8~60), Ud の回路規模は Uo の 24%~42% (頂点数 8~30) であった. 図 3 には, XC2V1000 FPGA の回路規模も示されている. XC2V1000 と同程度のスライス数で, Ud なら頂点数 30, Kd なら頂点数 60 まで扱えることが解る.

平均実行時間を図 4 に示す. ソフトウェア S の実行時間は, 頂点数の増加に伴って急激に増加する. 頂点数 14 以上では, Kd と Ud の実行時間は生成時間まで含めても S より短い.

4. 評 価

本章では, 3 章の結果を確認するため, 頂点数 16 の Kd と Ud を現実の FPGA に搭載して性能を評価した. 実装には Xilinx Virtex-II XC2V1000 FPGA を搭載した Insight MicroBlaze Development Kit を使用した. 設計・実装環境は 3 章で述べた通りである. FPGA ボードに搭載されたオシレータ (24 MHz) をクロックに用いるため, 25 MHz 動作という制約で論理合成~配置配線を実行した. データ依存回路の終了信号を FPGA のピンに出力し, ホスト側からパラレル I/O ボード経由で終了信号をポーリングして, 実行時間を測定した.

実装結果を表 1 に示す. 表に示された結果は, いずれも 100 組の入力グラフセット (3 章と同じもの) の平均である. Kd も Ud も, ソフトウェア S より短時間で実行を終了することが確認された. 回路の動作時間だけを見ると Ud は Kd より短い, 回路生成時間は Ud の方が長い, 総実行時間も Ud の方が長くなる. Kd の性能は S の 13.3 倍, Ud は 9.4 倍である. このとき Ud の論理規模は Kd の 2.2 倍であった.

参 考 文 献

- 1) S.Ichikawa, S.Yamamoto: Data dependent circuit for subgraph isomorphism problem, IE-ICE Transactions on Information and Systems, Vol.E86-D, No.5, (2003). (to appear)
- 2) 市川周一, ラターナセンタン ウドーン, 小西幸治: 部分グラフ同型判定アルゴリズムの FPGA による実装と評価, 情報処理学会論文誌 ハイパフォーマンスコンピューティングシステム, Vol.41, No.SIG5, pp.39-49(2000).