

修士学位論文要約（令和4年3月）

CMOS インバーティブルロジックの論理設計環境構築に関する研究

加藤 諒

指導教員：羽生 貴弘， 研究指導教員：鬼沢 直哉

A Study on building logic design environment for CMOS invertible logic

Makoto KATO

Supervisor: Takahiro HANYU, Research Advisor: Naoya ONIZAWA

Recently, it has been focusing on that CMOS invertible logic (CIL) which can operate bi-directionally has a potential advantage for a solution of inverse problem such as machine learning and factorization, while design tools as well as software libraries are not prepared to automatically generate the target CIL configuration. The proposed logic-design environment can automatically generate Hamiltonians from the given logic circuit information (written by the standard hardware description language Verilog-HDL). And a simulator in this environment is developed with C++ and OpenMP, a parallel processing API. The simulator. As a result, its simulation speed has increased 150 times and more than that of Python-based one. Furthermore, using this environment, we have investigated the effects of design parameters on the operating speed and quality of CIL.

1. はじめに

順方向と逆方向 2 つのモードで動作するインバーティブルロジック[1]を特殊なデバイスを使用せず、CMOS で実装する CMOS インバーティブルロジック (CIL)が近年研究されている[2]。CIL を用いた双方向演算は機械学習や逆問題を始めとして様々な分野での応用が期待できる。CIL である機能を実現する際には、その機能に対応するエネルギー関数(ハミルトニアン)を決定する必要がある。従来手法ではこのハミルトニアン決定に制約があり、特に大規模化に問題があった。そこで、本研究では設計、検証、分析の 3 つの面から CIL 研究を支援する論理設計環境を立ち上げた。

2. CIL 自動設計ツールの開発

提案設計手法[3]は、1)所望関数を Verilog-HDL などのハードウェア記述言語を用いて記述、2)論理合成

器を用いてゲートレベルネットリスト生成、3)ネットリストとテストベンチを開発ツールに入力、5)テストベンチの記述に従ってシミュレーションというフロー(図.1)からなる。CIL の論理設計は CIL のエネルギー関数であるハミルトニアンを係数を求めることによって行われる。所望の関数がある関数の合成で表現されるとき、その関数に

表.1 回路変換例

Function	# of node	E _{min}	E+1	E+1	Verification
Multiplier (input 4-bit)	97	-202	-94	40	✓
Multiplier (input 8-bit)	299	-652	-366	50	✓
Multiplier (input 16-bit)	1033	-2410	-1508	18	✓
Multiplier (input 32-bit)	3618	-8473	-6689	-3459	✓
Adder (input 4-bit)	38	-55	-21	21	✓
Adder (input 8-bit)	44	-35	-33	-35	✓
Adder (input 16-bit)	84	-67	-65	-67	✓
Adder (input 32-bit)	164	-131	-129	-131	✓
3x3 Matrix prod. (input 8-bit)	2814	-6270	-3684	48	✓

対応するハミルトニアン同士の単純な合成操作によって所望の関数に対応するハミルトニアンを求める事ができる[4]。この性質を用いて、各論理ゲートに対応するハミルトニアンを事前に求めておくことで、それらを組み合わせて作られる関数(組み合わせ論理回路)の CIL ハミルトニアンが得られる。

表に提案手法を用いた回路変換例を示す。Verilog-HDL を用いた RTL 設計から Design Compiler(Synopsys.Inc)を用いてネットリストを生成し、提案手法で CIL ハミルトニアンへと変換した。いずれの場合も問題なくハミルトニアンに変換できており、後述の自動検証も行っている。数千ノードからなる大規模なハミルトニアンも高速に求めることができた。

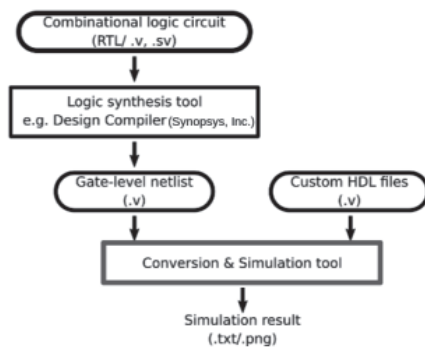


図.1 提案フロー

3. 高速シミュレーターの開発

大規模化による問題点はハミルトニアン設計だけでなく、シミュレーションにも存在する。CIL は一般的な FPGA などを用いたハードウェア実装が可能であるが、様々な所望関数、パラメータを柔軟に切り替えるの研究にはソフトウェアシミュレーションが適している。また CIL の制御は現在未知な要素が多く、観測すべきデータが多く存在する。そこで本研究では、C++と並列化 API, OpenMP を用いた高速シミュレーション機能と、その結果の可視化機能、及び回路の

表.2 シミュレーターの速度比較

シミュレーター/並列数	実行速度
シミュレーター 1(従来)/なし	1245 s
シミュレーター 2(提案)/1	8 s
シミュレーター 2(提案)/10	2 s

検証機能を有するツールを開発した。

Python を用いたシミュレーションと比較して、実行時間が150分の1となり、また、並列化によって10並列時に更に4分の1となった。並列化による高速化が10分の1で無いのは、並列化不能部分によるボトルネックによるものであると考えられる。また、ノード数が十分に多い時(数千程度)は並列化が効果的であるが、少ない時(数百程度)ではスレッドの生成に時間がかかるため、と1スレッドで実行時より低速となることがわかった。

4. ハミルトニアンの違いがCILの動作に与える影響

CIL は確率的に動作するため、解に到達するまでに時間を要し、また、正しい解に到達するとは限らない。大規模化によってこの求解時間、解の品質は一般に悪化する。デジタル回路の設計では遅延や電力等、制約条件から最適な論理設計が行われる。

デジタル回路において、あるゲート出力が駆動するゲート入力の数に Fanout という。この Fanout は CIL ハミルトニアンへと変換した時、ひとつのノードあたりのエッジ接続数の指標となる。論理合成器に与えるパラメータの調整によって Fanout の増加すなわちエッジ接続数の増加が CIL 乗算器の求解時間(解とのエラーが 0 になるまでのサイクル数)に与える影響を調査した。

調査の結果を図2に示す。縦軸が求解時間、横軸が回路中の最大 Fanout である。Max Fanout=2 のときよりも4のほうが早く解が求まり、そこから Fanout が大きくなるにつれ求解時間が増加する傾向が見て取れる。以上の結果より、CIL には各回路に最適な Fanout が存在すると考えられる。デジタル回路において Fanout を小さくするための戦略として、インバー

タなどが追加される事がある。Fanout=2 の時に求解時間が大きな理由として、インバータ等の追加によって CIL のノード数が増加し、状態の数が増えたことが考えられる。また、Fanout を大きくすることでノードの数が減る一方で、平均接続数は増加する。接続数の大きなノードは状態が変化しづらくなることが考えられ、局所解に落ちた場合、現在の制御方法では探索が十分に行われていない可能性がある。

5. まとめ

CIL 研究を支援する論理設計環境構築に関する研究を行い、ハードウェア記述言語を用いたハミルトニアン設計手法の考案を行い、これを実現するツールの実装を行った。また、並列化技術を用いた CIL

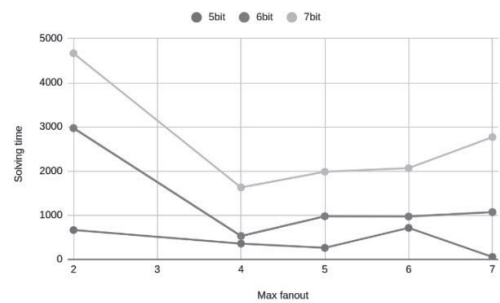


図.2CIL 乗算器における Max

シミュレーターの開発を行った。さらにこれらを使用して設計した算術演算 CIL について、設計パラメータがその動作に与える影響を調査し、パラメータ調整による CIL の高性能化の可能性を示した。

謝辞

本研究はキャノンメディカルシステムズ株式会社の協力のもと行われました。

文献

- [1] Kerem Camsari, Rafatul Faria, Brian Sutton, and Supriyo Datta. Physical Review X, 7, July 2017.
- [2] S. C. Smithson, N. Onizawa, B. H. Meyer, W. J. Gross, and T. Hanyu. IEEE Transactions on Circuits and Systems I: Regular Papers, 66(6):2263–2274, 2019.
- [4] M. Kato, N. Onizawa, and T. Hanyu, Journal of Applied Logics, vol. 8, no. 5 pp. 1311-1333, June 2021
- [3] J. D. Whitfield, M. Faccin, and J. D. Biamonte.. Europhysics Letters, 99(5):57004,2012.