様　式　Ｃ－１９、Ｆ－１９、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

平成 ２８ 年　 ６ 月　 ９ 日現在

機関番号：１５２０１
研究種目：基盤研究(C)（一般）
研究期間：2013～2015
課題番号：２５３３００６１

研究課題名（和文）機械学習を用いたカバレッジ駆動型ハードウェア検証の効率化に関する研究

研究課題名（英文）Improving Coverage Driven Verification for Hardware Using Machine Learning

研究代表者
浜口　清治（Hamaguchi, Kiyoharu）

島根大学・総合理工学研究科（研究院）・教授

研究者番号：８０２３８０５５

交付決定額（研究期間全体）：（直接経費）　　3,000,000円

研究成果の概要（和文）：マイコンやFPGAなどの設計工程では，製造前に行われる設計検証の工程の比率が大きく，効率化が課題である．シミュレーションによって設計検証を行う際，カバレッジと呼ばれる量的基準の増加を目指すのがカバレッジ駆動検証である．本研究では，ベイジアンネットワークと呼ばれる確率モデルに対する機械学習を利用して，より早くカバレッジを増加させるようなシミュレーション用のパターンを生成する．信号線の変化に着目したシミュレーションパラメータに対する機械学習および自動調整により，既存手法よりも短い時間でカバレッジを向上できることを示した．

研究成果の概要（英文）：In design process of microcomputers or FPGA devices, more efficient design verification is demanding. In simulation-based verification, the method for improving a quantitative metric called coverage is called coverage driven verification. In this research we uses machine learning for Bayesian networks for generating input patterns for simulation. Focusing on simulation parameters for the transition of signals in the design, our proposed method shows that coverage can be improved more quickly than the other existing methods.

研究分野：ハードウェアの設計支援技術

キーワード：シミュレーション検証　SATソルバ　カバレッジ駆動型検証　ベイジアンネットワーク

１．研究開始当初の背景

SoC(システム・オン・チップ)，マイコン，FPGA などの設計工程では，設計検証工程が 50%-80% と依然大きな比重を占めており効率化が課題となっている．シミュレーションベースの検証が広く用いられているが，人手で各機能のテストパタンを生成する手法に加えて，ランダムにテストパタンを生成して検証カバレッジを向上させる手法が採用されている．設計中に種々の条件に基づくカバーポイントを設定し，テストパタンによって各カバーポイントでその条件が満たされるかどうか,すなわち,カバーポイントがカバーされるかどうかを量的に評価する．カバレッジを増加させることを目標としてテストパタンの生成を行う方式はカバレッジ駆動型検証と呼ばれ広く用いられている．カバレッジの１つであるトグルカバレッジでは，１つの信号線のトグル(値の０から１および１から０への反転)がカバレッジポイントと呼ばれ，シミュレーションによって回路内のすべてのカバレッジポイントがカバーされた(すべての信号線で反転が起こった)とき 100%となる．

カバレッジ型検証に関しては，機械学習を利用してランダムパタン生成のパラメータを制御する手法があり，自動化という点で有効である．ランダムにテストパタンを生成する場合には，各入力信号線がとる値について，確率分布を定めるパラメータを設定する必要がある．機械学習を用いてこれらのパラメータを決定する手法は，リグレション検証(誤り修正後再び行う検証)での利用が考えられる．まず，誤り修正前に行ったランダムテストの結果から，パラメータの値と各カバレッジポイントがカバーされる割合を学習しておく．次に，リグレション検証では各カバレッジポイントがカバーされやすいパラメータの最適値を機械学習の結果から推定し，次々設定していって，カバレッジを増加させる．

機械学習の利用については，ベイジアンネットワークを用いた手法や帰納的プログラミングを用いた手法が提案されている．最も成功しているのは Intel 社の A. Ziv らによるベイジアンネットワークの機械学習によるアプローチである．ベイジアンネットワークは，複数の確率変数間の依存関係を有向グラフで表現したものであり，各ノードが確率変数１つに対応する．ランダムパタン生成では，各ノードにパラメータおよびカバレッジポイントが対応づけられる． 機械学習ではグラフ構造を固定して，データ集合から確率変数間の確率分布を学習する場合と，さらにデータ集合に最も適合するグラフ構造を自動的に推定する場合とがある．

しかし，既存手法は，マイクロプロセッサなど限られた設計への適用例しかなく，また，機械学習のアルゴリズムをそのまま転用しており，設計やカバレッジの特性を利用していないため，十分な性能を引き出し切れていると言えない．また，実際の設計では，例外処理などのように，ランダムテストでは生成確率が極微であるが，検証必須の機能が含まれている．学習用のデータ(シミュレーション時のテストパタン)が生成されない場合，機械学習のアルゴリズムを単純に適用するアプローチでは対応できない．これらは，機械学習を用いたカバレッジ駆動型検証手法の弱点であり，その克服が必要な状況にあった．

２．研究の目的

以上の状況を踏まえて，次の点を本研究の当初の目的とした．

(1) ベイジアンネットワークのグラフ構造を,設計記述から取り出す手法について検討・実装する．

入力信号に対する確率パラメータに対応するノードと，カバレッジポイントに対応するノードの間の依存関係を単純にグラフ構造で表現すると，カバレッジポイントがほぼすべての入力信号に依存するような意味のない構造が得られてしまう可能性が高い．回路をグラフとして分割することにより，大域的な構造を残したグラフを生成する手法などが考えられ，これらについて比較・検討を行って実装する．

(2) SAT ソルバの利用法について検討・実装する．

ランダムテストによって，発生しにくい条件に関しては，たとえば，ある特定の信号線が 1 にならない，といった形で観測することができる．この場合，SAT(satisfiability) ソルバによって，1 になるようなパタンを発見する．入力パタン１つでは不十分であるが,ダイバース SAT と呼ばれる，ハミング距離ができるだけ離れたパタンを，できるだけ多数生成する手法を用いることによって，確率パラメータとの対応づけを行うことができると考える．

(3) リグレション検証を行った場合の効果を実験的に評価する．

３．研究の方法

以下では，リグレッション検証を前提として，トグルカバレッジに着目したカバレッジ駆動型検証を考える．

(1) 機械学習によるカバレッジ駆動型検証

機械学習を用いたカバレッジ駆動型検証では，まず，シミュレーションパラメータの値の種々の組み合わせについて，シミュレーションを行う．その結果についてカバレッジ解析を行って，シミュレーションパラメータとカバレッジとの関係を機械学習で学習する．次に学習結果から，カバーポイントをカバーしやすいシミュレーションパラメータを推定して，これを用いてシミュレーションを行う．この手順を示したものが図１である．

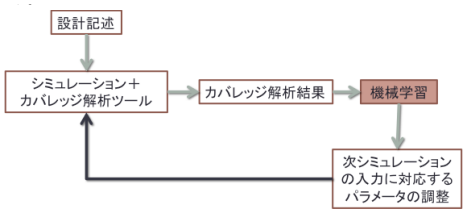


図 1：機械学習を用いたカバレッジ駆動型検証

ベイジアンネットワークはノードで表す複数の確率変数間の関係をグラフ構造で表し，因果関係を条件付き確率によって記述するグラフィカルモデルのひとつである．複数の変数間の関係は非巡回有向グラフで表される．例を図 2 に示す．グラフの構造に応じて，確率変数間に条件付き確率分布が与えられる．

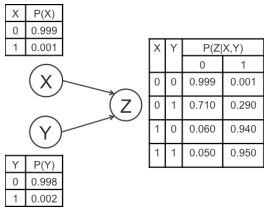


図 2．ベイジアンネットワークの例

ベイジアンネットワークの学習では，各確率変数に対応する観測値データをベクトル形式に集めたデータセットを与えて，このデータセットに対する学習スコアを最適にするようなベイジアンネットワークの構造と条件付き確率分布を求めることになる．

一部の確率変数が特定の値をとったとき，その他の確率変数の確率分布を求めることを確率推論という．ここでは，周辺分布の計算をもって確率推論とする．

ベイジアンネットワークによる機械学習をカバレッジ駆動検証におけるシミュレーションパラメータの調整に利用する手順は次の通りである．

まず，確率分布を数種類準備して，信号線ごとに確率分布を選択して，それに基づいてシミュレーションを行う．その際カバーされたカバーポイントを調べて，データセットを作成する．図 3 にシミュレーションフローを，図 4 にその結果得られるデータセットを示している．

次にこのデータセットに対して機械学習を行う．ベイジアンネットワークの構造を決めるアルゴリズムには，PC アルゴリズム，グリーディサーチ，Chow-Liu Tree 法，Rebane-Pearl Polytree 法などが知られている．

次に得られたベイジアンネットワークについて，カバーポイントに対応する確率変数に対して，カバーに相当する値を与えて，入力変数に対応する確率変数について確率推論を行う．その様子を図 5 に示す．

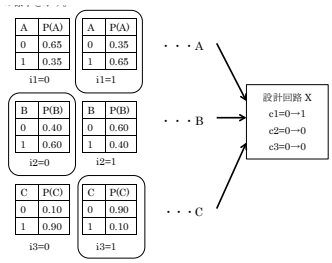


図 3：シミュレーションフロー

i1	i2	i3	c1	c2	c3
0	1	0	1	0	0

図 4：データセット（1 行分）

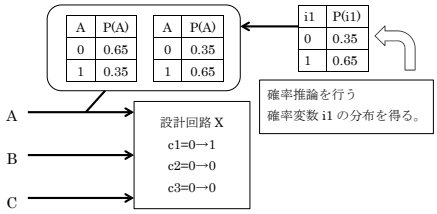


図 5：確率推論によるパラメータの設定

以上がベイジアンネットワークを用いたカバレッジ駆動型検証の流れとなる．

ベイジアンネットワークの構造を回路の構造から生成する手法については，4.(1)で述べるように予備的な実験を行った結果，良好な結果が得られなかった．このため，4.(2)および(3) で述べるように，異なるアプローチをとることとした．

(2) SAT ソルバを利用したテストパタン生成

ランダムシミュレーションをもちいる範囲では，簡単にはカバーすることができないカバーポイントが残存する．このようなカバーポイントに対しては，学習データがないため，機械学習は用いることはできない．これに対して，SAT ソルバを用いてテストパタンを生成する方式が考えられる．

SAT ソルバは与えられた論理式に対して，式が 1 となるような論理変数の割当を探索し，存在する場合はその割当(解)を出力し，存在しない場合は存在しないことを示すアルゴリズムまたはソフトウェアのことである．SAT ソルバを用いる方式では，カバーに関する論理条件(トグルカバレッジの場合は信号線の変化)と設計記述をすべて論理式で表現して，SAT ソルバを使って，カバーポイントをカバーするパタンを探索する．フローを図 6 に示す．

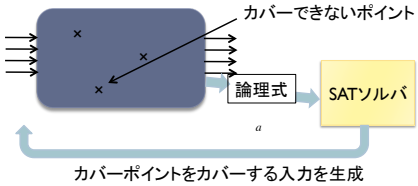


図 6：SAT ソルバを用いた検証フロー

設計は順序回路として与えられているため，論理式の生成にあたっては，多数のクロックサイクル数分のコピーを作ることになる．このため論理式は非常に大きなものとなり，SAT ソルバを用いた場合の時間コストは一般に大きい．SAT ソルバは通常 1 つの解のみを出力するが，SAT ソルバは動作中に探索空間について学習を行っており，解が存在しない条件を発見した場合，それを保持することにより探索を高速化している．このことから，一度 SAT ソルバを呼び出した際，複数の解を同時に発見することにより，効率の改善が期待できる．一方，単純に SAT ソルバに解を繰り返し生成させると，非常に似通った解を出力する傾向がある．

これに対して，本研究ではダイバース SAT ソルバを用いる．ダイバース SAT ソルバは，解の探索時に，その時点で発見している解からのハミング距離(解中の 0 および 1 の割り当てられ方が異なる変数の数)の総和が，可能な限り小さくなるように，変数への値の割当を選ぶ．このため，見つかった複数の解はターゲットとしているカバーポイントをカバーするという点では変わりはないが，互いに違いが大きいため，他のカバーポイントをカバーする能力が大きいことが期待できる．

ダイバース SAT ソルバを用いた場合のカバレッジ駆動型検証のフローを図 7 に示す．SAT ソルバのみでは，ランダムシミュレーションで比較的容易にカバーできるカバーポイントに対して，大きな時間コストをかけてテストパタンを生成してしまう可能性が大きいため，実際にはランダムシミュレーションと組み合わせて，検証を進める．

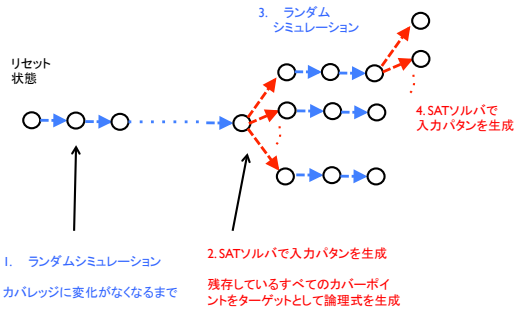


図 7：SAT ソルバを用いたカバレッジ駆動型検証

4．研究成果

(1) グラフ構造の抽出

3.(1)に示した手法のもとで，FIFO 回路および USB インターフェース回路に対して，入力とカバレッジポイントの数を限って，回路の構造を反映したグラフを人手で明示的に与えた場合と完全に機械学習に任せた場合で比較したところ，ほとんど変化しないか，悪化する場合もあることが判明した．これは，必ずしも元の因果関係に沿ったグラフ構造が，データセットに対して良いスコアを持つとは限らないことに起因していると予想された．このため方針を変更して，異なるアルゴリズムの利用やトグルカバレッジの性質に着目することとした．

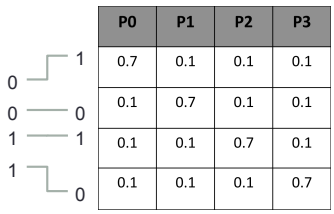
(2) 種々のアルゴリズムの適用

ベイジアンネットワークの学習アルゴリズムはいくつか知られている．3.(1) に示した手法のもと，種々のアルゴリズムの優劣を比較した．100 サイクルのシミュレーションを 100 回行った結果を使って学習を行った．同じ設計例に対して，各カバーポイントについて確率推論を行って，入力に対する確率分布を設定して，テストパタンを生成して，各カバーポイントのカバー回数を比較した．USB インターフェース回路に対する結果の一部を表 1 に示す．FIFO 回路や ADPCM 回路についても実験を行ったが，同様の結果となり，アルゴリズムによる差は確認できなかった．

表 1：USB インターフェース回路に対する結果

カバーポイント	rand	PC	Greedy	Rebane	Chow	NPC
c1_0	2701	3402	3402	3077	3247	3348
c1_1	2641	3011	3461	3144	3319	2613
c1_2	2694	2517	2688	2763	2692	2599
c1_3	2160	1886	1880	1886	1722	1798
c3_01	82	89	138	127	151	148
c3_10	85	87	121	118	109	128

(3) (2)の結果を踏まえて，構造学習のアルゴリズムは優劣がつかないことから，単純なヒルクライム法(グリーディヒューリスティック)による構造探索を採用した．一方，トグルカバレッジが対象であることから，1 サイクルに対する確率分布の他に，2 サイクルに対する確率分布を導入して計算機実験を行った．図 8 に使用した 2 サイクル系列に対する確率分布を示す．



P0	P1	P2	P3
0.7	0.1	0.1	0.1
0.1	0.7	0.1	0.1
0.1	0.1	0.7	0.1
0.1	0.1	0.1	0.7

図 8：2 サイクル系列に対する確率分布

IWLS2005 ベンチマーク回路から，100～6,000 ゲート規模の設計例に適用した結果を以下に示す．実験は Intel Core-i5 2.7GHz, 1GB のマシンを Linux Ubuntu 15.04 のもとで利用した．ベイジアンネットワークの学習には R のパッケージである bnlearn を，シミュレータは Icarus Verilog を使用した．1 回のシミュレーションは 250 サイクルとし，データセットを 100 個生成し，学習を行った．ランダムシミュレーションによるカバー率が 10%から 80% のカバーポイントをターゲットとしている．カバーポイント全てについて 1 回以上カバーするまでのシミュレーションの回数(1 回のシミュレーションは 250 サイクル)を比較した．結果を表 2 にまとめる．5 回繰り返した得られた回数の平均を示している．

表 2：カバーポイントをカバーするまでのシミュレーション回数

設計	ランダム	2サイクル系列	1サイクル系列
fifo	85	2	2
uart	100	1	2
simple-spi-top	8	8.3	12.4
usb_phy	4	3	14
tv80_core	20	18.9	34.1
s5378	8	6	57.6

これより，2 サイクル系列を用いた場合がシミュレーション回数の点で有利であることが明らかとなった．

(4) SAT ソルバを用いたテストパタン生成

3.(2) で説明した方式を実装して評価した．1 回のシミュレーションは 100 サイクルからなり，10 回続けてカバレッジの変化がなくなるまでランダムシミュレーションを行い，次にその最終状態を起点にして，ダイバース SAT ソルバで解を 50 個生成し，設計に入力する．これを実行時間の上限を定めて繰り返し行い，最終的なトグルカバレッジ(1 度でもカバーされたカバーポイントの割合)によって比較する．

設計例は IWLS2005 のベンチマークから 100～6,000 ゲート規模の設計を選んだ．Intel Core-i5 2.5GHz, 2GB のマシンを Linux Ubuntu 13.03 のもとで利用した．

結果を表 3 に示す．SAT の展開長は記述を何サイクル分コピーしたかを表している．これについては設計規模に合わせて手動で調整した．random はランダムシミュレーション，simpleSAT は単純な SAT ソルバの繰り返し，divSAT はダイバース SAT を用いた場合の結果である．SAT 実行時間と呼び出し回数はそれぞれ，SAT のみの実行時間の合計と SAT ソルバの呼び出し回数である．

表 3:SAT ソルバを用いたテストパタン生成

			random	simpleSAT			divSAT		
	SATの展開長	実行時間(秒)	Cov	Cov	SAT実行時間(秒)	呼出回数	Cov	SAT実行時間(秒)	呼出回数
usb_phy	200	1200	78.8%	97.8%	1126.05	6	100.0%	407.57	1
uart	250	1200	28.6%	94.6%	1065.03	17	100.0%	980.46	57
simple_spi_top	80	2000	96.7%	84.2%	1834.75	14	92.5%	1930.66	17
s5378.v	80	2000	92.0%	94.8%	1758.62	12	94.8%	1813.35	13
s13207.v	80	2000	59.1%	56.5%	1212.88	10	56.5%	1435.2	12
tv80_core.v	30	3200	94.2%	79.0%	2494.4	3	80.0%	3111.76	3

この表では，順に設計規模が大きくなっているが，大きい設計ほど SAT の展開長を大きくすることができない(SAT ソルバが解を見つけることができない)．このため，同じ実行時間ではランダムシミュレーションの方が高いカバレッジを示していると考えられる．usb_phy や uart は内部に制御が複雑なキュー構造を含んでおり，ランダムシミュレーションではカバレッジの改善が難しい．これらについては，SAT ソルバの利用が有効であると言える．

4.(2)および(3)の結果を総合すると，リグレッション検証での利用方法としては，まずランダムシミュレーションでカバーできるポイントについてデータを集めて，機械学習を行っておく．同時にカバーが難しいポイントに関しては，SAT ソルバでテストパタンを生成しておき，これらをリグレション検証の段階で組み合わせて用いることでカバレッジの改善を加速できると考えられる．

この際，SAT ソルバの負荷の軽減が必要であることが判明したが，ビットベクトルや数式なども扱うことができる SMT(Satisfiability Modulo Theory)ソルバの利用などを検討する必要がある．また，ベイジアンネットワークを利用するアプローチについて，当初良好な結果が得られなかったため，本研究の目標の一つであったクロスプロダクトカバレッジをターゲットとした検証については成果を得ることができなかった．一方，一般的な設計について，ベイジアンネットワークのアプローチにより，シミュレーション回数を削減できることが判明したため，クロスカバレッジや他のカバレッジ基準に対しても，同様の効果が得られると考えており，これらは今後の課題である．

５．主な発表論文等

〔学会発表〕(計 1 件)

浜口清治（発表者）：SAT ソルバを援用したカバレッジ駆動設計検証について，情報処理学会システムと LSI の設計技術研究発表会，2015 年 12 月 2 日，長崎県勤労福祉会館．

６．研究組織

(1)研究代表者

浜口 清治 (HAMAGUCHI KIYOHARU)

島根大学・大学院総合理工学研究科・教授

研究者番号：80238055