様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

# 科学研究費助成事業　　研究成果報告書

令和　３ 年　 ６ 月 １０ 日現在

機関番号：２５４０３
研究種目：基盤研究(C)（一般）
研究期間：2016～2020
課題番号：１６Ｋ００００８１

研究課題名（和文）深層学習を用いた配置配線手法の研究

研究課題名（英文）Research on place and route method using deep learning

研究代表者
弘中　哲夫（Hironaka, Tetsuo）

広島市立大学・情報科学研究科・教授

研究者番号：１０２５３４８６

交付決定額（研究期間全体）：（直接経費）　　3,500,000円

研究成果の概要（和文）：SA法による配置配線においてニューラルネットワークをコスト関数として適用する方法を確立した．適用にあたって単純に配置配線情報をニューラルネットワークに入力するだけでは，入力ノード数が非現実的な数になる．本研究では，配置と結線情報を配置位置と配線可能経路をマップ化することで入力データの意味を大きく失わずに大幅に入力ノード数を削減する方法を確立した．さらに同じネットリストから生成した2つの配置をニューラルネットに入力することでより良い配置を判定する手法を確立した．また，作成したニューラルネットワークをコスト関数とするSA法を従来の配置配線結果に適用して，さらに配置を改善することが可能となった．

研究成果の学術的意義や社会的意義
SA法のコスト関数は配線配置の可否を判定する数式モデルから作成している．そのため，性能向上を目指して再構成デバイスの構成を複雑にすると簡易な数式モデルで配線配置の可否を精度よく判定することが難しくなる．このコスト関数の精度が低下するとSA法を用いて生成される配置配線の品質が低下する．本研究では，深層学習を用いて訓練データである過去の配置配線結果から自動的にSA法のコスト関数を生成することの実現性を示した．これにより，精度の良い簡易な数式モデルが作成困難な場合においても，過去の配置配線結果を用いてより良いSA法のコスト関数を自動生成できる可能性を示した．

研究成果の概要（英文）：We revealed the method to apply a neural network as a cost function used in the SA method. If a neural network is simply applied, the placement and connection information will require an unrealistic number of input nodes. In this study, so we established a method to significantly reduce the number of input nodes without significantly losing the meaning of the input data by converting the placement and connection information to a map marked with placement position and the possibly routed area. Furthermore, we established a method to predict the better placement by comparing two placements generated from the same netlist by the neural network. Furthermore, it was also found that the placement performed by the conventional place and route methods can be improved by applying the SA method using the trained neural network as the cost function.

研究分野：コンピュータアーキテクチャ

キーワード：配置配線　ニューラルネットワーク　再構成可能デバイス

様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９（共通）

１．研究開始当初の背景

再構成可能デバイスは、一般に内部に多数用意した演算器、メモリ、論理演算素子を相互に接続する配線とスイッチでつながれた内部構成を持っている。そして、再構成可能デバイスの性能を引き出すためには、再構成可能デバイス上に実装するアプリケーションが要求する内部の演算器、メモリ、論理演算素子を最短の配線で接続できるように配置配線する必要がある。

この配置配線問題に対し、現在広く使われている再構成可能デバイスの代表である FPGA では、事前に生成したアプリケーション回路のネットリストを元に Simulated Annealing 法をベースとした VPR に代表される手法により、配線同士の干渉や、論理演算素子の過密をコスト関数によって抑えて論理演算素子を配置する手法を取っている。そして、その後にはダイクストラ法をベースとする配線手法である PathFinder に代表される手法で配線するといった配置後に配線するという 2 段構成の手法で配置を行った後に配線を行っている。このように 2 段階で配置を行ってから配線を行う方法では、配置を行う時に使用する配線同士の干渉や、論理演算素子の過密を事前に計算するコスト関数の善し悪しが鍵となる。

配線同士の干渉や、論理演算素子の過密を事前に見積もるコスト関数は一般に単純な構造を持つ再構成可能デバイスでは比較的簡単に作成でき、多数の研究が行われているが、近年の FPGA などの再構成可能デバイスで見られるように FPGA の性能向上のため、様々なサイズの多様なハード IP(メモリ、プロセッサ、乗算器、ALU 等々)を配置した場合、単純な方法で正確な見積を行う評価関数を作成する事が極めて困難になり、チップの面積効率を犠牲にして冗長な配線資源を用意する事で対応する必要がある。この事は実際の商用 FPGA でも配線の自由度を上げるため、デバイスの総チップ面積の 9 割以上を配線要素が占めているという報告がある事から確認できる。

このような状況から、もし、複雑な内部構造を持つ再構成可能デバイスにおいても正確に配線同士の干渉や、論理の過密を事前に計算するコスト関数が作成できれば、冗長に持つ配線資源を大幅に減らし、チップ面積当たりの性能が大幅に向上した再構成可能デバイスが実現可能になる。しかし、このようなコスト関数は個々の複雑な内部構造を持つ再構成可能デバイスの微妙なバリエーションで毎回異なる関数となり、統一的な手法で作成する事が困難である。

そこで、本研究ではこのような状況を鑑みて、複雑な内部構造を持つ再構成可能デバイスの配置配線用コスト関数を深層学習(Deep Learning)で配置配線に関する学習を行わせることで、配置配線用コスト関数となるニューラルネットワークを自動生成させる方法を検討する。

２．研究の目的

本研究は、再構成可能デバイスを効率良く使用する上で重要な論理演算素子や演算器等の IP の配置配線問題を深層学習(Deep Learning)したニューラルネットワークを援用することで解探索を行い、コンパクトで配線遅延の少ない配置配線を実現することを研究目的とする。この目的を達成するためには、次の 4 つの目標に取り組んだ。

（１）　配置配線問題を深層学習で取扱えるよう問題を定式化。
（２）　配置配線問題を深層学習するための学習データ自動生成の実現。
（３）　深層学習を行うために適切なニューラルネットワークアーキテクチャの開発。
（４）　高速な深層学習および、深層学習を行った結果を用いて高速に配置配線を実現する専用計算エンジンの開発。

以上を研究目的として研究を行った。

３．研究の方法

研究目的を実現するため、配置配線問題をニューラルネットワークで取扱えるよう問題定式化の研究と、配置配線問題をニューラルネットワークでの学習を可能にする学習データの自動生成法の研究を同時に行った。これは、学習データを作成できない方法で配置配線問題の定式化を行ってもニューラルネットワークを訓練できないからである。その後、ニューラルネットワークで正しく予測できる適切なニューラルネットワークアーキテクチャの評価研究を行った。

その後、高速なニューラルネットワークの学習および、学習を行った結果を用いて高速に配置配線を実現する専用計算エンジンの開発を行った。しかし、専用計算エンジンの開発については当初 FPGA を用い C ベース設計でエンジンを開発する予定であったが、ニューラルネットワークの学習とそれを用いた予測は、実際に行ってみた結果、FPGA で行う場合よりも GPU で行う方が学習においてはニューラルネットワークアーキテクチャの変更に対して高速かつ柔軟に対応できることが明らかになった。また、配置配線 CAD を構成する Simulated Annealing 法や配線を行う NC 法の実装を C ベース設計で FPGA に実装することを試みたが、研究開始当時の FPGA 付属の C ベース設計ツール Vivado HLS に対応しない C 言語記述などが多く、十分な高速化を達成するには時間が不足するため研究の本題である「深層学習を用いた配置配線手法の研究」を優先することとし、FPGA により計算高速化は断念して GPU 付き PC を用いて研究することとした。

４．研究成果

（１）配置配線問題をニューラルネットワークで取扱えるように定式化する研究

本研究では、そもそもニューラルネットワークをどのように扱えば、配置配線問題に適用でき

るかを研究する。研究を行うにあたって、比較的複雑な配置配線デバイスであると同時に均一な構造を持つのでモデル化しやすい再構成可能デバイスとして MPLD アーキテクチャを用いた。

配置配線を行う上でニューラルネットワーク単独で配置結果や配線を求めるのは学習可能な問題として実現することは困難である。そこで、本研究では、従来の配置配線における Simulated Annealing 法を用いる配置処理において使われるコスト関数をニューラルネットワークへ置き替えることを検討する。ニューラルネットワークのコスト関数への置き換えであれば、特定の配置に対するエネルギーをニューラルネットワークで求めることが可能である。

そして、従来の Simulated Annealing 法を用いる配置処理において、適切な配置配線結果が得られないのは、従来のコスト関数では次々と生成していく配置の近傍解の改善改悪が温度の高い時には適切に判断できても、最終段階の低温領域において判定が適切にできていないためである。これは、我々が MPLD アーキテクチャを用いて行った評価でも Simulated Annealing 法の終盤では、配置結果を実際に配線することで調査した結果、コスト関数により配置が改善されていると判定された配置の内で改善ができていた配置は平均 65%程度であった。つまり、Simulated Annealing 法を用いた配置処理の終盤において、近傍解配置の改善改悪判定が不正確なものになっている。これが、Simulated Annealing 法による最適化をよりいっそう進める上で障害になっている。そこで、本研究では学習により複雑なコスト関数が作成可能なニューラルネットワークを用いる。

Simulated Annealing 法で使用するコスト関数の入力は、論理演算素子の配置位置とそれらの論理演算素子間の接続関係、そして、その出力は配置の良し悪しに比例するエネルギーである。したがって、コスト関数と同等なものを出力できるようにニューラルネットワークを用いなければならない。しかし、N 個の論理演算素子の位置を表すビットマップと論理演算素子間の接続関係を示す N×N の行列というように単純な入力を用いると N が大きくなるとニューラルネットワークの入力層が非常に大きなものになり、計算量が膨大になる。そこで、配置を行う上で配置の良し悪しが判断できる概略情報をニューラルネットワークに入力することを基本方針とした。具体的には以下の方法を評価した。

- Congestion map：配線の混雑状況を重視した入力方法である。Congestion map は BoundingBox を用いて作成する。BoundingBox とは、接続関係のある論理演算素子同士を矩形で囲んだものである。Congestion map は BoundingBox で囲まれた部分全てに 1 を加算し、LUT が配置されている場所に 10 を加算して作成する。こうすることで 1 を加算した部分は配線の可能性があると認識させ、10 を加算した部分には論理演算素子が配置していると認識させることができると考えられる。図 1 を例に論理演算素子を LUT とすると、ネットリストの LUT1 と LUT3 は接続関係にある。この 2 つの LUT を BoundingBox で囲み、囲まれた部分全てに 1 を加算し、LUT1 と LUT3 の配置されている場所に 10 を加算する。複数の接続先がある場合は、全ての接続源と接続先を BoundingBox で囲み、同様に作成を行う。
- Manhattan map：配線の配線長を重視した入力方法である。Manhattan map はネットリストの接続関係と論理演算素子の配置情報を用いて作成する。まず、接続関係のある論理演算素子同士のマンハッタン距離を求める。マンハッタン距離の求め方は、接続源の論理演算素子の配置座標を(x1, y1)、接続先の論理演算素子の配置座標を(x2, y2)とすると、$|x1 - x2| + |y1 - y2|$で求める。図 2 を例に論理演算素子を LUT とすると、ネットリストの LUT1 と LUT3 は接続関係にある。接続源の LUT1 は(0, 3)、接続先の LUT3 は(1, 1)の位置に配置されている。この 2 つの LUT 間のマンハッタン距離は$|0 - 1| + |3 - 1| = 3$ となるので、(0, 3)に 3 を入力する。接続先が複数ある場合、その中で最大となる距離値を(x1, y1)に入力する。これをすべての接続関係について計算して Manhattan map を作成する。
- Manhatcon map：混雑度と配線長の両方を重視したマップである。Congestion map と Manhattan map の両マップを連結して作成する。

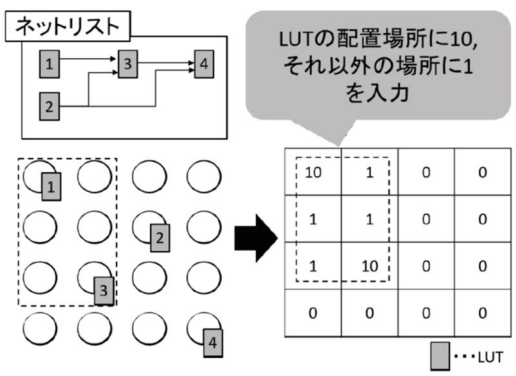


**図 1** Congestion map の作成法

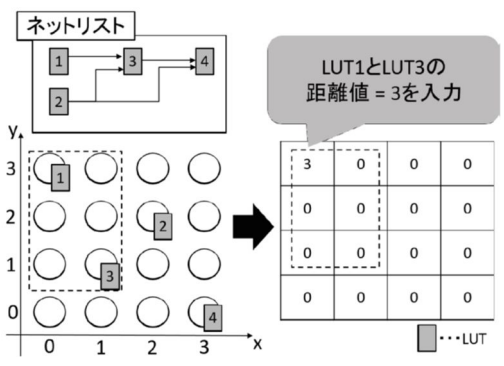


**図 2** Manhattan map の作成法

次に、コスト関数の出力は、配置の良さに比例した出力を得るのが理想的であるが、論理演算素子の配置の場合の数は、膨大な数になるのであらゆる場合についての一意の出力値を教師データとして事前に用意することは難しい。そこで、出力値を学習で作成することを検討した。具体的には、単独の配置から出力値を作るのではなく、図 3 に示すように 2 つの配置を比較し、より良い配置を選択する分類問題として訓練を行い、2 つの配置を正しく分類できるようになった後、2 つの出力の差が 2 つの配置の善し悪しの度合いで変化することを利用してその差を出力とした。なお、コスト関数として利用するとき、2 つの配置の片方を固定することで多様な配置に対する一様なコスト値となるようにした。

（2）学習データの自動生成法の研究

本研究では、ニューラルネットワークを訓練するための学習データ生成方法の研究を行った。学習データとなる配置配線結果を生成するための使用する回路のネットリストは、既存の回路から切り出したベンチマーク回路を用いるのが理想的であるが、回路規模や接続パターンのバリエーションなどに注目すると、多様な回路を収集するのが難しい。そこで、本研究では、平均論理演算素子数や平均接続数などのパラメータあらかじめ決めてランダムに回路を複数生成して学習データとして配置配線結果を生成するためのネットリストとして用いた。

次に学習データである配置配線結果を生成する方法として、次の２つの方法について検討した。

- 従来コスト関数を用いた手法：Simulated Annealing 法を実行するときに生成される配置の近傍解と、従来コスト関数から得たその配置の近傍解のコスト値を作成する。その後、得た配置のコスト値を用いて配置同士を比較し、2 つの配置とその良し悪し分類結果を作成した。
- 配置配線済みの結果から生成する手法：Simulated Annealing 法で配置配線を行った。その後、2 つの配置配線結果を配線に成功したか否か、総配線長の短い方はいずれかで比

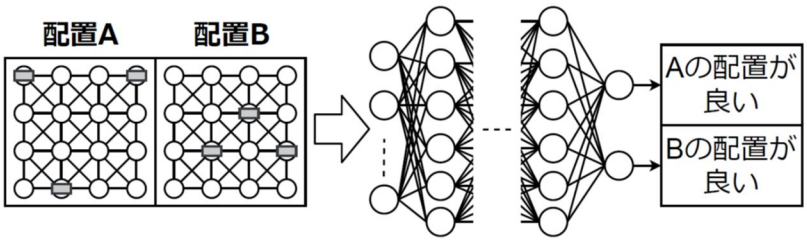


図 3　配置の良し悪しを比較するニューラルネットワーク

較し、2 つの配置とその良し悪し分類結果を作成した。

その結果、前者は非常に多くの学習データを配置が全く最適化されていない学習データから、配置が比較的最適化された学習データまで短時間で作成できるが、従来コスト関数の問題点である配置のコストが適切につけられていないという問題を継承する。そのため、この学習データで訓練したニューラルネットワークを Simulated Annealing 法に導入することで配置配線は可能であったが、どの結果も従来コスト関数を用いた Simulated Annealing 法を超える配置配線結果を得ることができなかった。それに対し、後者の配置配線済みの結果から生成する手法を用いた場合には、学習データが最適化された配置に大きく偏る。

（3）ニューラルネットワークアーキテクチャ

（2）の方法で作成した学習データを用いてニューラルネットワークのハイパーパラメータを変化させ、学習率と正解率を確認することで一番適切と思われるニューラルネットワークアーキテクチャを選択した。その結果、ニューラルネットワークアーキテクチャをそれぞれ、入力ノード×中間ノード×出力ノードの形式で示すと、manhattan は 2048×16×2、congestion は 2048×16×2、manhatcon は 4096×16×2 となった。

（4）Simulated Annealing 法に組み込んでの評価

（3）得た 3 つのニューラルネットワークを配置配線済み配置から作成した学習データを用いて訓練し、これをコスト関数として Simulated Annealing 法に導入した。その結果を評価したのが表 1 である。この結果を見ると多くの場合にそもそも配線できない配置となり、配線できる配置でも従来コスト関数を用いたものより配線成功率、平均総配線長がともに従来コスト関数を用いた結果より悪い、長いという配置配線結果が得られた。これは、単純にニューラルネットワークを Simulated Annealing 法のコスト関数として導入することに問題があることを示している。

表 1 のような結果になるのは、最適化が終わった配置配線済みの配置から作成した学習データでニューラルネットワークを訓練しているため、乱雑に置かれた初期配置や、最適化が進んでいない配置においてニューラルネットワークが適切に配置を分類できないためである。

表１ ニューラルネットワークのみをコスト関数として用いた場合

| | 従来コスト関数 | | NN入力形式manhattan | | NN入力形式congestion | | NN入力形式manhatcon | |
|---|---|---|---|---|---|---|---|---|
| 評価回路 | 配線成功率 (%) | 平均配線長 | 配線成功率 (%) | 平均配線長 | 配線成功率 (%) | 平均配線長 | 配線成功率 (%) | 平均配線長 |
| b1 | 100% | 55.8 | 97% | 223.1 | 60% | 128.4 | 97% | 232.4 |
| cm150a | 57% | 530.2 | 8% | 1175.9 | 0% | - | 2% | 1429.5 |
| cm42a | 80% | 251.1 | 13% | 551.3 | 2% | 445.5 | 2% | 537.0 |
| cm85a | 99% | 342.2 | 97% | 804.4 | 92% | 828.0 | 92% | 869.5 |
| cmb | 99% | 420.2 | 99% | 1004.5 | 94% | 1130.1 | 98% | 1182.7 |
| con1 | 100% | 132.8 | 100% | 366.5 | 98% | 292.0 | 100% | 377.9 |
| cu | 90% | 491.0 | 40% | 1091.4 | 18% | 1234.9 | 16% | 1302.9 |
| decod | 8% | 416.4 | 0% | - | 0% | - | 0% | - |
| i1 | 100% | 369.3 | 93% | 1062.7 | 89% | 1192.5 | 87% | 1242.6 |
| misex1 | 23% | 552.2 | 2% | 1127.0 | 0% | - | 0% | - |
| misex2 | 28% | 993.5 | 1% | 2118.0 | 0% | - | 0% | - |
| mux | 53% | 518.4 | 4% | 1188.0 | 0% | - | 3% | 1410.0 |
| parity | 100% | 155.6 | 100% | 519.6 | 98% | 464.3 | 100% | 539.7 |
| pm1 | 81% | 392.4 | 21% | 1010.4 | 5% | 1081.2 | 8% | 1119.1 |
| rd53 | 32% | 309.9 | 2% | 645.5 | 2% | 610.5 | 0% | - |
| sqrt8 | 29% | 725.0 | 3% | 1434.3 | 0% | - | 0% | - |
| x2 | 74% | 460.3 | 11% | 977.9 | 1% | 1052.0 | 2% | 1223.5 |

※評価は ODINⅡベンチマーク回路から 17 種類用いて評価を行っている。

そこで、Simulated Annealing 法を行うとき、初期配置から従来コスト関数での最適化がほぼ終わるまで従来コスト関数を用い、その後にニューラルネットワークによるコスト関数に切り替えて最適化を行った。このようにすることでニューラルネットワークの訓練ができなかった初期配置から従来コスト関数で有効な最適化ができる範囲までは、従来コスト関数を用いた最適化を行うことができる。さらにその後、配置配線後の配置データで訓練したニューラルネットをコスト関数として Simulated Annealing 法を行うことで従来コスト関数ではできなかったさらなる最適化を行うことができる。このような方法で最適化した結果を示したものが表 2 である。

表 2 を見ると、従来コスト関数とニューラルネットワークへの入力形式 congestion を組み合わせたものは、従来コスト関数より悪い結果が得られたが、従来コスト関数とニューラルネットワークへの入力形式 manhattan を組み合わせたもの、従来コスト関数とニューラルネットワークへの入力形式 manhatcon を組み合わせたものでは、優れているという評価結果を得た。なお、表 2 の各ニューラルネットワークアーキテクチャは表 1 同じである。

表 2 従来コスト関数とニューラルネットワークを組み合わせてコスト関数として用いた場合

| | 従来コスト関数 | | 従来コスト関数 & NN入力形式manhattan | | 従来コスト関数 & NN入力形式congestion | | 従来コスト関数 & NN入力形式manhatcon | |
|---|---|---|---|---|---|---|---|---|
| 評価回路 | 配線成功率 (%) | 平均配線長 | 配線成功率 (%) | 平均配線長 | 配線成功率 (%) | 平均配線長 | 配線成功率 (%) | 平均配線長 |
| b1 | 100% | 55.8 | 98% | 58.4 | 64% | 154.3 | 92% | 110.4 |
| cm150a | 57% | 530.2 | 62% | 493.5 | 4% | 1188.0 | 71% | 453.6 |
| cm42a | 80% | 251.1 | 73% | 229.8 | 2% | 541.0 | 59% | 218.0 |
| cm85a | 99% | 342.2 | 100% | 332.8 | 87% | 870.7 | 98% | 328.9 |
| cmb | 99% | 420.2 | 100% | 386.0 | 98% | 1133.7 | 99% | 418.4 |
| con1 | 100% | 132.8 | 100% | 118.0 | 100% | 332.7 | 100% | 176.4 |
| cu | 90% | 491.0 | 87% | 426.1 | 23% | 1220.2 | 77% | 397.5 |
| decod | 8% | 416.4 | 2% | 379.5 | 0% | - | 17% | 381.1 |
| i1 | 100% | 369.3 | 100% | 310.5 | 84% | 1206.9 | 99% | 354.0 |
| misex1 | 23% | 552.2 | 40% | 508.0 | 1% | 510.0 | 41% | 477.0 |
| misex2 | 28% | 993.5 | 26% | 851.8 | 2% | 784.0 | 22% | 803.4 |
| mux | 53% | 518.4 | 66% | 469.4 | 2% | 868.5 | 71% | 449.3 |
| parity | 100% | 155.6 | 100% | 164.3 | 99% | 516.6 | 100% | 245.0 |
| pm1 | 81% | 392.4 | 81% | 350.7 | 3% | 1095.7 | 88% | 322.7 |
| rd53 | 32% | 309.9 | 40% | 268.0 | 2% | 454.5 | 34% | 275.2 |
| sqrt8 | 29% | 725.0 | 46% | 628.2 | 2% | 1155.5 | 43% | 600.4 |
| x2 | 74% | 460.3 | 79% | 420.1 | 2% | 722.0 | 76% | 399.8 |

※評価は ODINⅡベンチマーク回路から 17 種類用いて評価を行っている。

## （5）まとめと今後の課題

以上の結果から、ニューラルネットワークを単独で、従来のコスト関数の代わりに用いることは難しい。しかし、従来コスト関数を用いた Simulated Annealing 法で最適化した配置をニューラルネットワークコスト関数に用いてさらに最適化することで、従来の配置手法よりさらに配置を最適化できる可能性を示すことができた。

今後はより大規模で複雑な再構成可能デバイスに適用できるように、再構成可能デバイスの大規模化にともなって増加するニューラルネットワークの学習時間の削減に取り組んでいきたい。

５．主な発表論文等

〔雑誌論文〕　計5件（うち査読付論文　0件／うち国際共著　0件／うちオープンアクセス　0件）

| | |
|---|---|
| １．著者名<br>夏目 優一、鎌田 時生、窪田 昌史、谷川 一哉、弘中 哲夫 | ４．巻<br>120 |
| ２．論文標題<br>SA法のコスト関数として配置品質判定ニューラルネットワークを用いた 再構成可能デバイスの配置アルゴリズムの提案 | ５．発行年<br>2020年 |
| ３．雑誌名<br>信学技報 リコンフィギャラブルシステム研究会 | ６．最初と最後の頁<br>71-76 |
| 掲載論文のDOI（デジタルオブジェクト識別子）<br>なし | 査読の有無<br>無 |
| オープンアクセス<br>オープンアクセスではない、又はオープンアクセスが困難 | 国際共著<br>－ |

| | |
|---|---|
| １．著者名<br>鎌田 時生, 窪田 昌史, 谷川 一哉, 弘中 哲夫 | ４．巻<br>119 |
| ２．論文標題<br>コスト関数にニューラルネットワークを導入した論理素子配置アルゴリズムの検討 | ５．発行年<br>2019年 |
| ３．雑誌名<br>信学技報 リコンフィギャラブルシステム研究会 | ６．最初と最後の頁<br>13-18 |
| 掲載論文のDOI（デジタルオブジェクト識別子）<br>なし | 査読の有無<br>無 |
| オープンアクセス<br>オープンアクセスではない、又はオープンアクセスが困難 | 国際共著<br>－ |

| | |
|---|---|
| １．著者名<br>藤石 秀仁, 鎌田 時生, 弘中 哲夫, 谷川 一哉, 窪田 昌史 | ４．巻<br>118(334) |
| ２．論文標題<br>細粒度再構成可能デバイスMPLDにおけるディープラーニングを用いた論理素子配置の良し悪し判定 | ５．発行年<br>2018年 |
| ３．雑誌名<br>信学技報 リコンフィギャラブルシステム研究会 | ６．最初と最後の頁<br>71-76 |
| 掲載論文のDOI（デジタルオブジェクト識別子）<br>なし | 査読の有無<br>無 |
| オープンアクセス<br>オープンアクセスではない、又はオープンアクセスが困難 | 国際共著<br>－ |

| | |
|---|---|
| １．著者名<br>山下 裕司, 窪田 昌史, 谷川 一哉, 弘中 哲夫 | ４．巻<br>118(215) |
| ２．論文標題<br>FPGAの配置配線結果を使用したMPLDの配置配線ツールの検討 | ５．発行年<br>2018年 |
| ３．雑誌名<br>信学技報 リコンフィギャラブルシステム研究会 | ６．最初と最後の頁<br>61-66 |
| 掲載論文のDOI（デジタルオブジェクト識別子）<br>なし | 査読の有無<br>無 |
| オープンアクセス<br>オープンアクセスではない、又はオープンアクセスが困難 | 国際共著<br>－ |

| | |
|---|---|
| １．著者名<br>田中智大，谷川一哉，弘中哲夫，石黒隆 | ４．巻<br>116 |
| ２．論文標題<br>細粒度再構成可能デバイスMPLDのIOを考慮したSOMベース配置手法 | ５．発行年<br>2016年 |
| ３．雑誌名<br>信学技報 リコンフィギャラブルシステム研究会 | ６．最初と最後の頁<br>29-34 |
| 掲載論文のDOI（デジタルオブジェクト識別子）<br>なし | 査読の有無<br>無 |
| オープンアクセス<br>オープンアクセスではない、又はオープンアクセスが困難 | 国際共著<br>－ |

〔学会発表〕　計10件（うち招待講演　0件／うち国際学会　3件）

| |
|---|
| １．発表者名<br>夏目 優一、鎌田 時生、窪田 昌史、谷川 一哉、弘中 哲夫 |
| ２．発表標題<br>SA法のコスト関数として配置品質判定ニューラルネットワークを用いた 再構成可能デバイスの配置アルゴリズムの提案 |
| ３．学会等名<br>電子情報通信学会李コンフィギャラブルシステム研究会 |
| ４．発表年<br>2020年 |

| |
|---|
| １．発表者名<br>Tokio Kamada, Atsushi Kubota, Tetsuo Hironaka |
| ２．発表標題<br>Study of logic element placement algorithm which introduced neural network to cost function |
| ３．学会等名<br>The International Conference for High Performance Computing, Networking, Storage and Analysis（国際学会） |
| ４．発表年<br>2019年 |

| |
|---|
| １．発表者名<br>鎌田 時生, 窪田 昌史, 谷川 一哉, 弘中 哲夫 |
| ２．発表標題<br>コスト関数にニューラルネットワークを導入した論理素子配置アルゴリズムの検討 |
| ３．学会等名<br>電子情報通信学会リコンフィギャラブルシステム研究会 |
| ４．発表年<br>2019年 |

１．発表者名
Ryota Miyauchi, Akira Kojima, Hideyuki Kawabata, and Tetsuo Hironaka

２．発表標題
A Study of a Parallel Architecture for Accelerating Batch-Learning Self-Organizing Map by using Dedicated Hardware

３．学会等名
34th International Technical Conference on Circuits / Systems, Computers and Communications (ITC-CSCC 2019)（国際学会）

４．発表年
2019年

１．発表者名
Yuji Yamashita, Atsushi Kubota, Kazuya Tanigawa and Tetsuo Hironaka

２．発表標題
Reconfigurable Device MPLD and The CAD tool

３．学会等名
The International Conference for High Performance Computing, Networking, Storage and Analysis（国際学会）

４．発表年
2017年

１．発表者名
谷川 一哉 , 弘中 哲夫

２．発表標題
再構成可能デバイスMPLD/SePLDにおける設計アルゴリズムについて

３．学会等名
DAシンポジウム2017

４．発表年
2017年

１．発表者名
荒瀬郁実，窪田昌史，谷川一哉，弘中哲夫

２．発表標題
細粒度再構成可能デバイスMPLDの論理セル配置問題を対象としたSA法における迷路法を用いたコスト算出方法

３．学会等名
電子情報通信学会李コンフィギャラブルシステム研究会

４．発表年
2017年

１．発表者名
荒瀬郁実，田中智大，谷川一哉，弘中哲夫，窪田昌史，石黒隆

２．発表標題
細粒度再構成可能デバイスMPLDの論理セル配置問題を対象としたSA法におけるコスト関数改善の検討

３．学会等名
第18回IEEE広島支部学生シンポジウム

４．発表年
2016年

１．発表者名
田中智大，谷川一哉，弘中哲夫，石黒隆

２．発表標題
細粒度再構成可能デバイスMPLDのSOMベース配置手法へのリスト型データ構造導入による高速化の検討

３．学会等名
第18回IEEE広島支部学生シンポジウム

４．発表年
2016年

１．発表者名
田中智大，谷川一哉，弘中哲夫，石黒隆

２．発表標題
細粒度再構成可能デバイスの配置配線問題におけるニューラルネットワークを用いた配置配線手法

３．学会等名
LSIとシステムのワークショップ2016

４．発表年
2016年

〔図書〕　計0件

〔産業財産権〕

〔その他〕

―

６．研究組織

| | 氏名<br>（ローマ字氏名）<br>（研究者番号） | 所属研究機関・部局・職<br>（機関番号） | 備考 |
|---|---|---|---|

７．科研費を使用して開催した国際研究集会

〔国際研究集会〕　計0件

８．本研究に関連して実施した国際共同研究の実施状況

| 共同研究相手国 | 相手方研究機関 |
| --- | --- |