

ILPを用いたPAE向け最適クラスタリングによる グローバル配線の最小化

寺田 蒼生[†] 吉田 実由[†] 岩崎 凌大[†] 飯田 全広^{††}

[†] 熊本大学 大学院自然科学教育部 〒860-0855 熊本県熊本市中央区黒髪 2-39-1

^{††} 熊本大学 半導体・デジタル研究教育機構 〒860-0855 熊本県熊本市中央区黒髪 2-39-1

E-mail: [†]{264d8847,259d8860,266d9401}@st.kumamoto-u.ac.jp, ^{††}iida@cs.kumamoto-u.ac.jp

あらまし eFPGA (embedded FPGA) IP の新しい論理セルとして提案されている PAE Cell は、多出力かつ LUT と比較して少ない構成メモリ数を実現できる利点を持つ。一方で、その多出力構造ゆえにクラスタ内のフィードバックが増加し、LCB(Local Connection Block) の面積が増大するという課題がある。これに対し、先行研究では配線資源を制限したスパースクロスバー構造が提案され、その有効性が示されている。しかしクラスタリング手法は先行研究での構造探索用の手法に留まっており、その構造に対しての標準的な EDA ツールである VPR のアルゴリズムの有効性は十分に検証されていない。本研究ではグラフマッチングと ILP (整数線形計画法) を使用し現実的な計算時間で準最適解を求める手法を確立し、既存手法の妥当性を評価する。MCNC ベンチマーク回路による評価の結果、比較的良好なパッキング性能を示した VPR と提案手法の間においても、外部入力数が約 5.9%、外部出力数が約 2.9% の差があることを確認した。これにより VPR のアルゴリズムは PAE Cell に対して一定の有効性を示すものの、改善の余地があることが実証された。

キーワード クラスタリング, ILP, eFPGA, PAE Cell

Global Routing Minimization via ILP-Based Optimal Clustering for PAE

Soi TERADA[†], Miyu YOSHIDA[†], Ryo IWASAKI[†], and Masahiro IIDA^{††}

[†] Kumamoto University Graduate School of Science and Technology

2-39-1 Kurokami, Chuo-ku, Kumamoto, 860-0855 Japan

^{††} Kumamoto University Research and Education Institute for Semiconductors and Informatics

2-39-1 Kurokami, Chuo-ku, Kumamoto, 860-0855, Japan

E-mail: [†]{264d8847,259d8860,266d9401}@st.kumamoto-u.ac.jp, ^{††}iida@cs.kumamoto-u.ac.jp

Abstract Programmable AND Element (PAE) cells reduce configuration memory in eFPGA IPs by supporting multiple outputs more efficiently than conventional LUTs. However, this structure increases intra-cluster feedback, expanding the Local Connection Block (LCB) area. Although previous research proposed a sparse crossbar structure to address this, existing clustering techniques remain limited to architectural exploration, and the effectiveness of the standard VPR algorithm for PAE has not been fully verified. In this study, we establish a methodology using graph matching and Integer Linear Programming (ILP) to derive near-optimal solutions efficiently and evaluate existing methods. MCNC benchmark evaluations show that the proposed method reduces external inputs by 5.9% and outputs by 2.9% compared to VPR. These results demonstrate that while VPR exhibits some effectiveness for PAE, there remains room for optimization.

Key words Clustering, ILP, eFPGA, PAE Cell

1. はじめに

eFPGA(embedded FPGA) IP は、ASIC や SoC に組み込み可能な FPGA 回路ブロックである。これを用いることで、製造後でもチップ内の機能変更や修正が可能となり、設計の柔軟性が

向上する。しかし、論理合成可能な eFPGA IP では、小規模な SRAM を高密度に実装することが構造的に難しいため、構成情報の保持には SRAM の約 6 倍の面積を占有する D-FF を代用する場合が多く、これがチップ面積を圧迫する要因となっている [1]。この面積課題を解決するために PAE (Programmable

AND Element) Cell [2] が提案された。これは、従来の LUT と比較して構成メモリ数を大幅に削減できる特長を持つ。一般的な LUT では、入力数を N とすると真理値表の保持に 2^N 個のメモリを必要とし、指数関数的に増大する。これに対し PAE Cell では、必要な構成メモリ数は入力数 N にほぼ比例する形となる。例えば 4 入力の場合、LUT の 16 ビットに対し PAE Cell は 8 ビットと、同等の論理表現能力を維持しつつメモリ量を半減している。

また、PAE Cell は複数の出力を容易に取り出せる多出力構造を有しており、回路全体の論理セル数を削減できる利点がある。しかし、この多出力構造はクラスタ内の LCB (Local Connection Block) におけるフィードバック線の増加を招き、結果として配線ブロックの面積を増大させる課題が生じている。この問題に対し、フルクロスバーの柔軟性を制限して構成メモリを削減する PAE Cell 向けのスパースクロスバー構造が提案された [3]。

しかしながら、PAE Cell とスパースクロスバーの組合せを対象としたクラスタリング手法は未成熟であり、VPR 等の既存アルゴリズムの有効性は十分に検証されていない。特に、PAE Cell のような多出力構造はグローバル配線の増加を招きやすく、クラスタリング手法がこの構造的特徴に適切に対応できない場合、配線リソースの増大に伴ってチップ面積の拡大を招く。そこで本研究では既存手法の妥当性を評価する手法を確立し、定量的評価を通じて将来的な最適化に向けた指針を得ることを目的とする。

2. 先行研究

2.1 PAE Cell [2]

PAE Cell の構成要素である PA (Programmable AND) 回路を図 1 に示す。PA 回路は 2 入力 AND ゲートを基本演算素子とし、その前段および後段に構成メモリ (M_0, M_1) によって制御される反転機能を付加した構造を有する。具体的には、AND ゲートの一方の入力側に M_0 制御の XOR ゲートが配置され、入力信号の反転・非反転を選択できる。同様に、出力段にも M_1 制御の XOR ゲートが挿入されており、出力信号の反転・非反転を制御する。例えば、 $M_0 = 1$ の場合は入力信号 y が反転されて AND ゲートに入力され、 $M_1 = 0$ の場合は AND ゲートの出力がそのまま出力される。図 2 に、3 個の PA 回路と 4 個のマルチプレクサ (MUX)、さらに接続切り替え用の構成メモリ M_6, M_7 を組み合わせた 4 入力 3 出力の PAE Cell (4-3-PAE) の構造を示す。この M_6, M_7 を用いることで PA 回路同士の接続関係を動的に切り替えることが可能である。1 つの PAE Cell は、2 ビットの構成メモリを含む PA 回路 3 つと接続制御用の 2 ビット、計 8 ビットで構成される。同じ 4 入力を持つ 4-LUT が 16 ビットの構成メモリを必要とすることに対し、PAE Cell は論理セル単体でのメモリ数を半減できる利点がある。一方で、論理実装能力においては、PA 回路の組合せと多出力構造を活かすことで、従来の LUT と同等以上の柔軟性を確保している。

2.2 スパースクロスバー構造

PAE Cell は多出力構造を持つため、クラスタ内のフィードバック信号の本数が LUT 方式よりも多くなる。具体的には、4

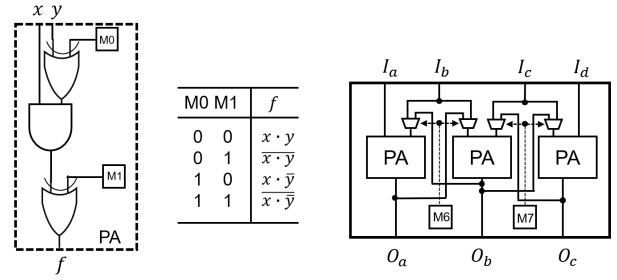


図 1 PA 回路の構造と機能 図 2 4 入力 3 出力 PAE Cell の構造

個の論理セルを格納するクラスタにおいて、LUT 方式のフィードバックが 4 本であるのに対し、4-3-PAE では 12 本に達する。この影響で入力 MUX の規模が拡大し、論理セル自体の構成メモリを削減できても配線部のメモリが増加して面積削減効果が薄れてしまう恐れがある。

そこで文献 [3] では、PAE 向けに最適化されたスパースクロスバー構造が提案されている。PAE Cell は物理的なゲートの接続構造を直接再現しているため、LUT のように入力の順序を任意に入れ替えることが困難である。そのため、既存の LUT 向け構造をそのまま転用するのではなく、PAE の接続特性を考慮した構造探索が行われた。

同文献では、クラスタの外部入力数 N を 16 または 12 とした際の比較を行うとともに、構成メモリ削減のために PAE の各入力ポート (MUX) の入力を 8 本に制限した構造を検討している。フルクロスバーを用いた場合、MUX への入力は外部入力 12 本、フィードバック信号 12 本の計 24 本となり、構成メモリは $\lceil \log_2 24 \rceil = 5$ ビット必要になるのに対し、スパースクロスバーでは入力を 8 本に限定することで、構成メモリは $\log_2 8 = 3$ ビットとなる。具体的には、8 本の入力内訳である外部入力接続数 E とフィードバック数 F ($E + F = 8$) の比率を固定し、現実的な配線制約のもと各ポートの接続候補をランダムに選択する手法によって構造の探索と評価が行われた。最適な配線能力を得る構成を明らかにするため、 F を 6 から 1 まで変化させた複数のパターンについて実験がなされている。

4-LUT および 4-3-PAE それぞれのフル/スパースクロスバー構造 (計 4 種) を対象に、論理セルと配線構造の両面から比較した結果、4-3-PAE のスパースクロスバーが最少の構成メモリ数を達成した。この構造は、 $F = 6, E = 2$ の場合のみ例外的に削減率が下回ったものの、同セルのフルクロスバーに対し 14.4% 以上のメモリ削減を実現している。また一連の評価から、クラスタ外部入力 $N = 12$ 、MUX への外部入力本数 $E = 3$ 、フィードバック本数 $F = 5$ の組合せが、パッキング性能と構成メモリ削減率のバランスの観点から最適であると結論付けられている。図 3 にこの検討に基づく PAE 向けのスパースクロスバー構造を示す。図 3 では PAE 1 の I_a への入力のみを示している。

また、入力ポートの傾向に基づいた規則的な構造を採用することで、ランダム探索による最良構造と同等の削減率を再現できることが示されており、その構造を表 1 に示す。例えば、 P_1 のポート I_a には、外部入力の 1, 5, 9 番 ($E = 3$) と、クラスタ内

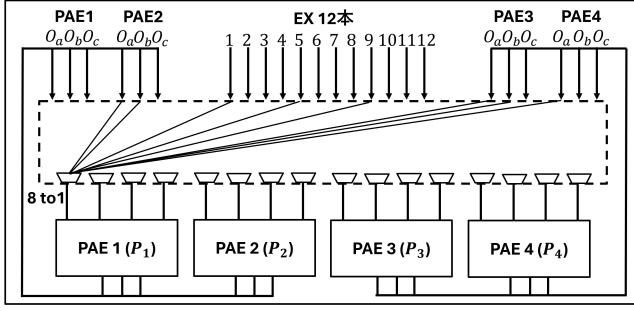


図3 PAE 向けのスパースクロスバー構造

表1 PAE ポート接続可能なリソース

PAE	ポート	外部入力 (ext.in)	他 PAE 出力 (O_a, O_b, O_c)
P_1	I_a	1, 5, 9	$P_2(O_a, O_b), P_3(O_a, O_b), P_4(O_c)$
	I_b	2, 6, 10	$P_3(O_a, O_b), P_4(O_a, O_b), P_2(O_c)$
	I_c	3, 7, 11	$P_4(O_a, O_b), P_2(O_a, O_b), P_3(O_c)$
	I_d	4, 8, 12	$P_2(O_a, O_b), P_3(O_b, O_c), P_4(O_c)$
P_2	I_a	5, 2, 11	$P_3(O_a, O_b), P_4(O_a, O_b), P_1(O_c)$
	I_b	6, 3, 12	$P_4(O_a, O_b), P_1(O_a, O_b), P_3(O_c)$
	I_c	7, 4, 9	$P_1(O_a, O_b), P_3(O_a, O_b), P_4(O_c)$
	I_d	8, 1, 10	$P_3(O_a, O_b), P_4(O_b, O_c), P_1(O_c)$
P_3	I_a	9, 6, 3	$P_4(O_a, O_b), P_1(O_a, O_b), P_2(O_c)$
	I_b	10, 7, 4	$P_1(O_a, O_b), P_2(O_a, O_b), P_4(O_c)$
	I_c	11, 8, 1	$P_2(O_a, O_b), P_4(O_a, O_b), P_1(O_c)$
	I_d	12, 5, 2	$P_4(O_a, O_b), P_1(O_a, O_b), P_2(O_c)$
P_4	I_a	4, 8, 10	$P_1(O_a, O_b), P_2(O_a, O_b), P_3(O_c)$
	I_b	1, 5, 11	$P_2(O_a, O_b), P_3(O_a, O_b), P_1(O_c)$
	I_c	2, 6, 12	$P_3(O_a, O_b), P_1(O_a, O_b), P_2(O_c)$
	I_d	3, 7, 9	$P_1(O_a, O_b), P_2(O_a, O_b), P_3(O_c)$

の他の PAE の出力である P_2 の O_a, O_b , P_3 の O_a, O_b , P_4 の O_c ($F = 5$) の計 8 本が入力候補として接続される構成となる。

3. クラスタリング手法

PAE Cell に対応したスパースクロスバー構造については、文献[3]において過度な配線自由度を下げ、構成メモリ削減の有効性が示されている。しかしそのクラスタリング手法については、構造探索用に作成されたものに留まっており、標準的な EDA ツールである VPR の適用可能性や、そのアルゴリズムの有効性は十分に検証されていなかった。本章では、既存手法のアルゴリズムと課題について述べ、その妥当性を検討する基礎とする。

3.1 AAPack [4]

VPR で使用されている AAPack というパッキングアルゴリズムについて説明する。AAPack は貪欲法 (Greedy Algorithm) をベースとしたヒューリスティックなパッキング手法であり、主に以下の手順でクラスタリングを実行する。まず未割り当ての論理ブロックの中から、最も多くの信号線が接続されているブロックを起点となるシードとして選択する。シードの決定後は、残りの候補ブロック p に対して、対象クラスタ B への親和性 (Affinity) を計算し、追加する論理ブロックを逐次的に決定する。候補ブロック p と生成過程にあるクラスタ内の論理ブ

ロック集合 B の間の親和性は (1) のように定義される。

$$Aff(p, B) = \frac{(1 - \alpha) \cdot nets(p, B) + \alpha \cdot connections(p, B)}{num_pins(p)} \quad (1)$$

ここで、 $nets(p, B)$ は p と B のブロック間で共有されている信号数を示し、 $num_pins(p)$ はブロック p が使用する総ピン数である。 α は共有ネットと外部接続の評価の重みを調整するパラメータである。また、 $connections(p, B)$ は B の外部への接続に関する指標であり、(2) のように定義される。

$$connections(p, B) = \frac{1}{ext(p, B) + packed(p) + 1} \quad (2)$$

$ext(p, B)$ はまだどのクラスタにも割り当てられていない外部ピンの数、 $packed(p)$ は既に他のクラスタに割り当てられたブロックへ接続されているピンの数を表す。これらの数式からわかるように、 $nets(p, B)$ が大きい (クラスタ内部での配線共有が多い) ほど親和性 Aff は高く評価される。一方で、外部への未割り当てピン $ext(p, B)$ や、他クラスタへの接続ピン $packed(p)$ が多いブロックは、 $connections(p, B)$ の値が小さくなるため、親和性は低く評価される。これにより、クラスタ外部への配線増加を抑制している。

また、これらの評価値を $num_pins(p)$ で除算 (正規化) することで、入力数の異なる多様な規模のブロック間で公平な比較と選択を可能としている。クラスタに論理ブロックを追加する際には、その割り当てがクラスタ構造の制約に対して物理的に可能かも同時に検証される。この過程を、クラスタにブロックが挿入できなくなるまで繰り返すことでパッキングを行う。

しかしながら、AAPack は貪欲法に基づくヒューリスティックな手法であり、すべての候補を探索できないため最適解の導出が困難である点や、LUT を前提とした設計が、多出力構造を持つ PAE Cell における複雑なフィードバック接続に十分に対応できているかが不明であるといった課題がある。

3.2 文献[3]のクラスタリング手法

文献[3]では、AAPack の枠組みをベースに、AAPack が逐次的なパッキングを行うのに対し、あらかじめ PAE の組合せを確定したうえでパッキング判定を行う。これは、フィードバック接続や入力共有関係をあらかじめ確定させることで、PAE Cell 特有の物理制約を確実に満たすことを目的としている。

具体的なアルゴリズムの実行手順としては、まず未クラスタの任意の PAE をシードとして選択し、直接的な信号接続や外部入力の共有度に基づき、関連性の強い近傍 PAE 群を「探索ウィンドウ」として抽出する。続いて、抽出されたウィンドウ内の PAE における全組合せを列挙し、各組合せがスパースクロスバー構造においてパッキング可能かどうかの判定を一括で行う。判定のプロセスでは、信号の共有状況やフィードバック接続の整合性を確認し、新規の外部入力が必要な場合でも、それが未使用の入力ピンリソースに割り当て可能であれば有効な候補として採用する。

最終的に、これら有効な候補の中から PAE の重複を避けつつ、最も多くの素子を集約できる組合せを選択し、対応する PAE をクラスタへと登録する。以上の工程をすべての PAE がクラスタに含まれるまで繰り返すことでパッキングを完了させる。

本手法はウィンドウ内での一括判定により接続整合性を保証している一方で、依然としてシード起点の局所探索に留まっている。そのため、チップ全体を通じたクラスタ数の最小化や外部配線リソースの効率といった広域的な最適性については、検証の余地が残されている。また、膨大なクラスタ候補の組合せを確定させてから、パッキングを行うというアプローチは、回路規模の増大に伴い処理時間が大幅に拡大する傾向にあり、実用における計算コストの抑制が課題として挙げられる。

4. ILP を用いたクラスタリング

先述の通り、AAPack は 1 出力の LUT を前提としたヒューリスティックな手法であり、多出力構造を持つ PAE Cell に対しては、LUT と同等の品質が得られるかは不明である。また、文献 [3] の手法についてもパッキング性能の妥当性の評価は行われていない。そのため、クラスタ構造の実装効率を客観的に評価するためには、既存ツールの探索特性に依存しない厳密解の導出が求められる。厳密解を得るアプローチとして、ILP（整数線形計画法）は有効である一方、回路内のすべての PAE の組合せを変数として扱くと、探索空間の増大により現実的な時間での求解が困難になる。

本研究では、この計算量に対処するため、ILP による最適化の前段としてグラフマッチングを用いたクラスタ候補の絞り込みを行い、ILP を用い準最適解を求める手法を提案する。図 4 に提案手法のフローを示す。本フローにおいて四角は処理ステップを、楕円は工程で得られる中間データまたは出力結果を表す。4.1 節ではグラフマッチングによる候補の列挙について述べ、4.2 節では得られた候補集合を用いた ILP の定式化について述べる。

4.1 グラフマッチング

本節では、クラスタ候補を限定し列挙するためのグラフマッチング手法について述べる。本手法では、PAE レベルネットリストから抽出した連結部分グラフと CLB 内部の配線可能性を表現したアーキテクチャグラフの 2 種類の有向グラフを取り扱い、回路側から抽出した連結部分グラフが、CLB のアーキテクチャグラフで被覆可能かを判定し、クラスタ候補を列挙する。図 5 に PAE の連結部分グラフの例、図 6 に CLB のアーキテクチャグラフの例を示す。この場合、図 5 の PAE の ① から ④ と図 6 の BLE 1 から 4 が対応している。

アーキテクチャグラフは、2.2 節で説明した先行研究において有効と評価されたクラスタ構造から作成したグラフを用いる。この際、構造上で実現可能なすべての内部接続を 1 つのアーキテクチャグラフ内にエッジとして定義する。

PAE レベルネットリストからの連結部分グラフの構築について説明する。まず、ネットリスト上の全 PAE をそれぞれ起点としてシードを選択する。各シードに対し、接続関係にある隣接 PAE を深さ優先探索 (DFS) によって再帰的に探索することで、

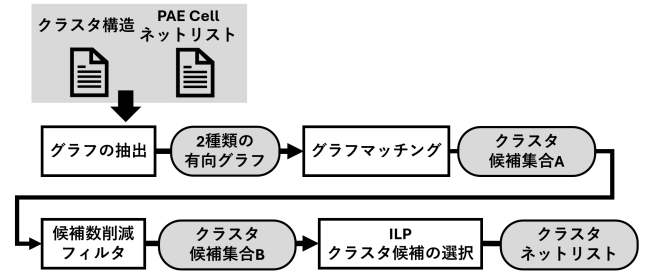


図 4 提案手法のフローチャート

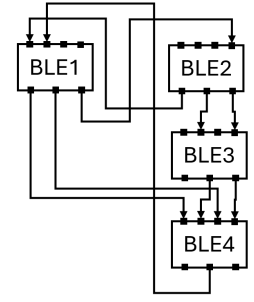
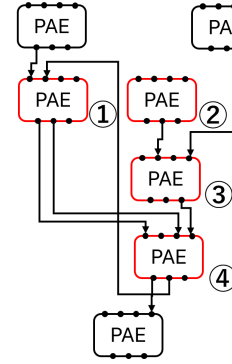


図 5 PAE の連結部分グラフ 図 6 CLB のアーキテクチャグラフ

PAE の組合せである連結部分グラフを抽出する。抽出する部分グラフのノード数は、クラスタのスロット数に基づき、2 から最大 4 の範囲とする。この際、探索における組合せ爆発を抑制するため、各サイズごとに候補数の上限を設け、候補の列挙を行う。この制約は、信号を内部化し外部配線を削減する効果が高い直接接続を優先するため妥当性をもつ一方で、共通入力をもつ非連結な組合せや回路全体での整合性に必要な解を排除するリスクを内包している。

次に、抽出された PAE の組合せが、スパースクロスバー構造に対してマッピング可能かを判定する。これは回路ネットリストから抽出された連結部分グラフとアーキテクチャグラフとをグラフマッチングを行うことで実現する。各 PAE がもつ入力ポート $I_a \sim I_d$ および出力ポート $O_a \sim O_c$ が、アーキテクチャ上で定義された接続リソースと一対一の対応が可能か検証する。具体的には図 5 の PAE の ① から ④ が、図 6 の BLE 1 から 4 とグラフが一致するかを確認する。このグラフマッチングを通じ、ポートの競合や配線が不可能な組合せは物理制約違反として候補から除外される。ここまでで図 4 のクラスタ候補集合 A が生成される。

最後に、外部からの接続要求を順次、利用可能な外部入力ポートへ割り当てを行う。ここで、すべての外部接続を空ポートに矛盾なく割り当てられた組合せのみを最終的なクラスタ候補とする。ここまでで図 4 のクラスタ候補集合 B が生成される。このように、あらかじめ物理制約を満たす候補を厳選しておくことで、全 PAE から無作為に 4 つを選ぶ場合の変数が $O(n^4)$ となるのに対し、本手法の探索空間は $O(n)$ のオーダーに抑えられる。これにより、後段の ILP における変数（組合せ数）を削減し、現実的な時間での求解を可能にしている。

4.2 ILP：クラスタ候補の決定

本節ではクラスタ候補から ILP を用いて最適解を求める方法について述べる．第 4.2 節の手順によって抽出されたクラスタ候補の集合から，回路全体を過不足なく被覆し，かつ外部接続数を最小化する最適な組合せを決定するため，整数計画法（ILP）を用いた定式化を行う．まず最適化モデルの構築にあたり以下の集合，定数および変数を定義する．

- P ：回路に含まれる全 PAE の集合
- C ：物理制約を満たす全クラスタ候補の集合
- $x_j \in \{0, 1\}$ ：候補 $j \in C$ を採用するか否かの二値変数（採用する場合に 1，そうでない場合に 0）
- w_j ：候補 j を採用した際に発生する外部配線数（候補内の総接続数から内部接続数を差し引いた値）
- ρ_j ：候補 j のサイズに応じたペナルティ係数

本手法の目的は，使用するクラスタ総数と外部配線総数の和を最小化することであり，目的関数 Z を次のように定義する．

$$\text{minimize } Z = \sum_{j \in C} (w_j + \rho_j) \cdot x_j \quad (3)$$

ここで，ペナルティ係数 ρ_j はクラスタ内の PAE 充填率を高めるために設定される．クラスタに含まれる PAE 数が少ない候補ほど高いペナルティを与え，最大サイズである 4 個に近い候補を優先的に採用することで，面積効率の向上を図る．本実験では，サイズ 1 の候補には 2.0，サイズ 2 には 0.5，サイズ 3 および 4 には 0.1 の重みを設定した．また，最適化の解において，全ての PAE がいずれか一つのクラスタに必ず，かつ重複なく属することを保証するため，以下の集合分割制約を課す．

$$\sum_{j \in C, p_i \in j} x_j = 1 \quad (\forall p_i \in P) \quad (4)$$

この制約により，全 PAE が一意のクラスタへ割り当てられ，論理的な矛盾や欠落のないパッキング結果が保証される．このようにして，候補を限定しているため最適解とは言えないが，最適解に近い値，準最適解を導出する．

5. 実験結果・評価

本章では，グラフマッチングおよび整数計画法（ILP）を用いて算出した準最適解を基準とし，既存のクラスタリング手法の性能評価を行う．本評価の目的は，算出した準最適解と既存手法のパッキング性能を定量的に比較・分析することで，PAE 向けスパースクロスバー構造に適したクラスタリング手法の指針を得ることである．

5.1 実験環境および条件

評価には，MCNC ベンチマーク回路の 20 回路（alu4, apex2, des, ex5p など）を使用する．アーキテクチャには 4-3-PAE を用い，第 2 章の表 1 で示した構成のスパースクロスバー構造を対象とする．比較対象は，先行研究である文献 [3] の手法，および VPR で標準的に採用されている AAPack の 2 種類とする．評価の公平性を期すため，すべての実験は同一の計算機環境で実施した．実験に使用した計算機の詳細を表 2 に示す．

表 2 実験環境

項目	内容
CPU	Intel Core i9-7900X
GPU	NVIDIA TITAN V
Memory	64 GB (DDR4 32GB × 2)

各ベンチマーク回路に対し，各手法で 10 回のパッキングを行い，その中で得られた最良の結果を採用する．これは，初期シードの選択やアルゴリズム内のヒューリスティックな要素に起因する結果のばらつきを排除し，各手法の最大性能を正しく評価するためである．

5.2 評価指標

本実験では，以下の 4 つの指標を用いてパッキング性能を評価する．表 3 に評価指標を示す．

表 3 パッキング性能の評価指標

指標	説明
外部入力総数（Ext.In）	全クラスタの外部入力信号ポートの総数
外部出力総数（Ext.Out）	全クラスタの外部出力信号ポートの総数
クラスタ数	回路の論理機能を実装するために要したクラスタ（CLB）数
実行時間（s）	パッキング処理の完了に要した時間

なお，本研究の提案手法および文献 [3] の手法については，クラスタ候補数の削減に要した時間も処理時間として実行時間を含めるものとする．

5.3 結果

表 4 に，各ベンチマーク回路におけるクラスタ数，外部入力本数，外部出力本数，および実行時間の評価結果を示す．表 4 の（）内には提案手法で求めた準最適解との増減率を示す．表 4 より，既存の標準ツールである AAPack は，提案手法による準最適解と比較して，クラスタ数において平均-3.3% という結果を示した．これは，AAPack が論理ブロックをクラスタ内に高密度に充填する能力に優れており，回路全体をより少ないクラスタ数で収容できていることを意味する．また文献 [3] については，平均 0.9% 増加しており，提案手法の方がより効率的に回路全体を収容できていることがわかる．

しかし，外部入力本数および外部出力本数に着目すると，AAPack は提案手法と比較してそれぞれ平均 5.9%，および 2.9% 増加していることが確認された．これは，AAPack のパッキングアルゴリズムが局所的な探索かつ貪欲法であるため，クラスタ内の密度を高める一方で，グローバル配線を最小化できていないことを示唆している．また文献 [3] については，提案手法と比較してそれぞれ平均 23.9%，および 35.9% 増加しており，こちらもグローバル配線の削減において課題が残る結果となった．

次に，パッキング完了までに要した実行時間について評価を行う．表 4 が示す通り，AAPack はヒューリスティックな貪欲法を採用しているため，大規模なベンチマーク回路においても数秒から数十秒程度で処理を完了しており，提案手法と比較して平均 98.6% 短い．また文献 [3] については，提案手法と比較して平均 12.7% 短い結果となったが，実用上の実行時間として

表 4 パッキング結果の比較

回路名	クラスタ数			外部入力本数			外部出力本数			実行時間 [s]		
	提案手法	文献 [3]	AAPack	提案手法	文献 [3]	AAPack	提案手法	文献 [3]	AAPack	提案手法	文献 [3]	AAPack
alu4	183	196 (7.1%)	189 (3.3%)	1,816	2,181 (20.1%)	1,934 (6.5%)	790	1,083 (37.1%)	851 (7.7%)	108.9	215.8 (98.2%)	2.2 (-98.0%)
apex2	145	157 (8.3%)	150 (3.4%)	1,382	1,709 (23.7%)	1,470 (6.4%)	654	901 (37.8%)	701 (7.2%)	81.7	186.0 (127.7%)	2.1 (-97.4%)
apex4	175	188 (7.4%)	182 (4.0%)	1,819	2,146 (18.0%)	1,908 (4.9%)	905	1,145 (26.5%)	948 (4.8%)	155.9	192.7 (23.6%)	2.2 (-98.6%)
bigkey	405	374 (-7.7%)	349 (-13.8%)	3,181	3,824 (20.2%)	3,259 (2.5%)	935	1,497 (60.1%)	1,061 (13.5%)	340.8	406.2 (19.2%)	2.6 (-99.2%)
clma	646	648 (0.3%)	629 (-2.6%)	5,874	7,167 (22.0%)	6,243 (6.3%)	2,763	3,654 (32.2%)	2,719 (-1.6%)	679.6	639.4 (-5.9%)	20.6 (-97.0%)
des	395	354 (-10.4%)	339 (-14.2%)	3,255	3,847 (18.2%)	3,356 (3.1%)	1,272	1,755 (38.0%)	1,342 (5.5%)	235.1	360.0 (53.1%)	2.8 (-98.8%)
diffeq	267	270 (1.1%)	259 (-3.0%)	2,225	2,862 (28.6%)	2,430 (9.2%)	895	1,199 (34.0%)	928 (3.7%)	162.2	249.3 (53.6%)	4.6 (-97.2%)
dsip	348	346 (-0.6%)	324 (-6.9%)	3,133	3,778 (20.6%)	3,261 (4.1%)	932	1,518 (62.9%)	1,083 (16.2%)	795.5	467.5 (-41.2%)	4.4 (-99.5%)
elliptic	664	684 (3.0%)	622 (-6.3%)	5,847	7,511 (28.5%)	6,189 (5.8%)	2,140	2,776 (29.7%)	2,123 (-0.8%)	972.8	915.0 (-5.9%)	14.6 (-98.5%)
ex1010	549	599 (9.1%)	569 (3.6%)	5,694	6,765 (18.8%)	5,936 (4.3%)	2,652	3,437 (29.6%)	2,748 (3.6%)	657.9	776.6 (18.0%)	3.0 (-99.5%)
ex5p	113	121 (7.1%)	118 (4.4%)	1,088	1,324 (21.7%)	1,173 (7.8%)	595	736 (23.7%)	631 (6.1%)	63.7	109.7 (72.2%)	2.2 (-96.6%)
frisc	668	691 (3.4%)	655 (-1.9%)	6,027	7,572 (25.6%)	6,442 (6.9%)	2,106	3,002 (42.5%)	2,207 (4.8%)	783.9	683.0 (-12.9%)	16.5 (-97.9%)
misex3	146	157 (7.5%)	153 (4.8%)	1,425	1,732 (21.5%)	1,519 (6.6%)	661	884 (33.7%)	714 (8.0%)	76.9	170.0 (121.1%)	2.2 (-97.2%)
pdc	447	482 (7.8%)	456 (2.0%)	4,464	5,396 (20.9%)	4,743 (6.3%)	2,086	2,752 (31.9%)	2,196 (5.3%)	397.7	574.4 (44.4%)	2.8 (-99.3%)
s298	183	193 (5.5%)	189 (3.3%)	1,827	2,162 (18.3%)	1,919 (5.0%)	746	1,046 (40.2%)	739 (-0.9%)	106.0	208.4 (96.7%)	3.9 (-96.3%)
s38417	1,065	996 (-6.5%)	975 (-3.8%)	7,968	10,104 (26.8%)	8,439 (5.8%)	3,401	4,741 (39.4%)	3,269 (-3.9%)	1,804.1	764.6 (-57.6%)	3.4 (-99.8%)
s38584.1	1,048	1,035 (-1.2%)	1,016 (-3.1%)	7,939	10,478 (32.0%)	8,524 (7.4%)	3,742	5,168 (38.1%)	3,773 (0.8%)	1,755.2	727.3 (-58.6%)	33.2 (-98.1%)
seq	162	173 (6.8%)	168 (3.7%)	1,581	1,935 (22.4%)	1,689 (6.8%)	756	1,013 (34.0%)	812 (7.4%)	94.6	173.0 (82.9%)	2.2 (-97.7%)
spla	369	398 (7.9%)	382 (3.5%)	3,651	4,398 (20.5%)	3,867 (5.9%)	1,686	2,240 (32.9%)	1,775 (5.3%)	277.4	437.8 (57.8%)	2.5 (-99.1%)
tseng	239	231 (-3.3%)	225 (-5.9%)	1,821	2,322 (27.5%)	1,990 (9.3%)	899	1,071 (19.1%)	891 (-0.9%)	142.7	204.7 (43.5%)	3.5 (-97.6%)
合計	8,217	8,293 (0.9%)	7,949 (-3.3%)	72,017	89,213 (23.9%)	76,291 (5.9%)	30,616	41,618 (35.9%)	31,511 (2.9%)	9,692.5	8,461.3 (-12.7%)	131.5 (-98.6%)

はまだ長く、アルゴリズムには改善の余地があると言える。

6. 考 察

実験結果より、提案手法と AAPack の間には性能上のトレードオフが確認された。クラスタ数に関しては、AAPack が提案手法を平均 3.3% 下回る結果となったが、これは既存手法の充填能力の高さを示すものである。一方で、外部入出力本数については AAPack が提案手法と比較して平均で入力 5.9%、出力 2.9% 上回っている。eFPGA の実装において、使用クラスタ数はデバイス容量内に収まっていれば、数 % の違いが問題になることは少ない。対照的に、外部配線数の増加はチップ面積の増加に直結する。したがって、クラスタ数の微増を許容してでも、面積増大の主要因となるグローバル配線の最小化を優先すべきであると考ええる。今後は、より厳密な検証を行うため、実際のチップ面積に基づく定量的な評価を行う必要がある。

一部回路において、AAPack の外部出力本数が提案手法よりも少なくなる場合が確認された。これはグラフマッチングで候補を限定する過程で、組合せ爆発を回避するために設けた候補数上限が影響していると考えられる。具体的には、s298 という回路を除き、この現象は比較的規模の大きい回路で発生している。回路規模の増大に伴い、有力な解が探索空間から除外されたことが、性能逆転の要因と推察される。この点は、提案手法における探索アルゴリズムの改善の余地を示すものである。

実行時間に関しては、提案手法と比較して AAPack が平均 98.6% と、実用面での優位性は AAPack にある。本研究で用いた ILP ベースの手法は、現実的な運用を想定した EDA ツールではなく、パッキング性能の理論的な上限を明らかにするための評価指標（基準値）としての役割を主眼としている。

文献 [3] については、実行時間を除くすべての項目において提案手法よりも各種指標が増加する結果となった。局所探索かつクラスタの充填率を重視した設計がこのような結果をもたら

したと考えられる。

これら知見に基づき、第一に AAPack 等の評価関数においてグローバル配線最小化の指標を強化し、シード選択方法を改良することで、スパースクロスバー構造への最適化を図ることが考えられる。これは独自プログラムを実用化する際にも同様の指針となる。第二に、本研究では外部配線数を評価指標としたが、これが最終的なチップ面積（配線領域）に与える影響をより詳細に検証し、クラスタ数とのトレードオフを評価する必要がある。第三に、提案手法をより厳密にするべく、分割統治法の適用による大規模回路への対応や、グラフマッチングにおいてより有効な解を優先的に残す工夫を導入し、理論的な上限値をより厳密に算出することを目指すたい。

謝 辞

本研究における提案手法の構築、評価にあたり、多大なるご協力をいただいた本研究室卒業生の平野大喜氏、および修士課程修了生の佐々木龍也氏に深く感謝いたします。

文 献

- [1] I.Kuon and J.Rose, "Measuring the Gap Between FPGAs and ASICs," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol.26, no.2, pp.203-215, Feb.2007, doi:10.1109/TCAD.2006.884574.
- [2] R. Iwasaki, K. Seto and M. Iida, "The PAE Cell: A Novel Multiple Outputs Logic Cell and Technology Mapping for eFPGA," 2025 IEEE Nordic Circuits and Systems Conference (NorCAS), Riga, Latvia, 2025, pp. 1-7, doi: 10.1109/NorCAS66540.2025.11231311.
- [3] 吉田実由・岩崎凌大・瀬戸謙修・飯田全広：PAE Cell に対応したスパースクロスバー構造の提案と評価, RECONF2025-45, pp. 122-127(2025)
- [4] J. Luu, J. Anderson, and J. Rose, "Architecture description and packing for logic blocks with hierarchy, modes and complex interconnect," in Proceedings of the 19th ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, pp. 227-236, Feb. 2011, doi: 10.1145/1950413.1950457.