

令和 6 年度
修士学位論文

セルフタイム型データ駆動プロセッサ搭載 FPGA ボードの電力推定法

A Study on Power Estimation of FPGA Board
Equipped With Self-Timed Data-Driven Processor

1275110 松坂 拓海

指導教員 岩田 誠

2025 年 2 月 28 日

高知工科大学大学院 工学研究科 基盤工学専攻
情報学コース

要 旨

セルフタイム型データ駆動プロセッサ搭載 FPGA ボードの電力 推定法

松坂 拓海

近年, IoT(Internet of Things) の技術進歩により様々な分野に IoT が浸透しており, その中で重要な役割を果たすのが IoT エッジデバイスである. IoT エッジデバイスは様々な応用の可能性を考慮し, 高い計算性能, 省電力性, 設計柔軟性が重要である. これに対して, セルフタイム型パイプライン STP で実現されたデータ駆動型プロセッサ DDP の FPGA 実装が有望である. DDP は並列処理性能に優れることから高い計算性能を持ち, 非同期回路であるため省電力性に優れる. FPGA は設計柔軟性に優れている上, 様々なラインナップがあるため目的に沿ったデバイスの選択ができる. FPGA の電力評価には, ベンダより商用 FPGA 設計ツール内で電力評価機能が提供されている. しかし, 非同期回路に基づく DDP を FPGA に実装する場合, 既存ツールは FPGA に特化しているため, 精度が低い. 加えて, 周辺回路を含めたボード全体の電力を評価できない問題がある.

そこで, 本研究では, 独自の DDP 搭載 FPGA ボード電力推定モデルに基づく電力推定手法を検討した. 独自の DDP 搭載 FPGA ボード電力推定モデルには DDP, STP, FPGA の特性を考慮する必要があるが, FPGA の電力特性は不明であり, 事前 to 取得する必要がある. その方法についても検討した. 提案手法を Digilent 社の FPGA ボード Zybo Z7-20 を対象に実装評価を実施した結果, 提案モデル A は平均絶対百分率誤差が 0.488%~3.297%, 提案モデル B は平均絶対百分率誤差が 0.973%~3.407%であることを確認した.

キーワード IoT エッジデバイス, STP(Self-timed Pipeline), DDP(Data-Driven Processor), FPGA(Field-Programmable Gate Array), 電力評価法, 電力推定モデル

Abstract

A Study on Power Estimation of FPGA Board Equipped With Self-Timed Data-Driven Processor

Takumi MATSUSAKA

In recent years, the advancement of IoT (Internet of Things) technology has led to its widespread adoption across various fields, with IoT edge devices playing a crucial role. For achieving higher potentiality in divers applications, IoT edge devices require high computational performance, low power consumption, and design flexibility. In this regard, FPGA implementation of a data-driven processor (DDP) realized with a self-timed pipeline (STP) is promising. The DDP offers high computational performance due to its excellent parallel processing capability and achieves low power consumption as it is based on asynchronous circuits. FPGA provides design reconfigurability and offers a wide range of product lineups, allowing for device selection according to specific needs.

For power evaluation of Synchronous circuits, commercial FPGA design tools provided by vendors include power estimation tools. However, when implementing an asynchronous circuit-based DDP on an FPGA, these tools, which are optimized for FPGAs, tend to have low accuracy. Additionally, they cannot evaluate the power consumption of the entire board, including peripheral circuits.

Therefore, this research examines a power estimation method based on a custom power estimation model for FPGA boards equipped with DDP. In this model, we utilize the characteristics of DDP, STP, and FPGA as heuristic knowledge. Since the power

characteristics of the FPGA are unknown, they must be obtained in advance. This study also investigates methods for acquiring these basic characteristics.

To evaluate the proposed method, an we used the Digilent FPGA board Zybo Z7-20 to implement the DDPs. The results showed that the proposed Model A had a mean absolute percentage error (MAPE) of 0.488%~3.297%, while the proposed Model B achieved a MAPE of 0.973%~3.407%. These findings confirm that the proposed power estimation model enables power estimation with a reasonable level of accuracy.

key words IoT edge devices, self-timed pipelines (STP), data-driven processor (DDP), FPGA(Filed-Programmable Gate Array), power estimation method, power estimation model

目次

第 1 章	序論	1
1.1	研究背景	1
1.2	提案する DDP 搭載 FPGA ボードの電力評価手法	3
第 2 章	データ駆動型搭載 FPGA ボードを構成する要素と電力評価方法の検討	4
2.1	緒言	4
2.2	セルフタイム型パイプライン STP	4
2.3	データ駆動型プロセッサ DDP	5
2.3.1	Mstage	5
2.3.2	MMCAMstage	6
2.3.3	MMRAMstage	7
2.3.4	PSstage	7
2.3.5	FPstage	8
2.3.6	MAstage	8
2.3.7	COPYstage	8
2.3.8	Bstage	8
2.4	FPGA ボード	9
2.5	DDP 搭載 FPGA ボードの電力評価方法の検討	11
2.5.1	デバイス電力の実測方法	11
2.5.2	デバイス電力の推定方法	12
2.6	結言	13
第 3 章	DDP 搭載 FPGA ボードの電力推定法	15
3.1	緒言	15

目次

3.2	DDP 搭載 FPGA ボードの電力推定モデルの検討	15
3.2.1	回路構成	16
3.2.2	動作周波数	17
3.2.3	動作確率	20
3.2.4	FPGA デバイスの電力特性	23
3.3	DDP 搭載 FPGA ボードの電力推定法	24
3.3.1	DDP 搭載 FPGA ボード電力推定モデル A	25
3.3.2	DDP 搭載 FPGA ボード電力推定モデル B	25
3.4	結言	26
第 4 章	実装評価	27
4.1	緒言	27
4.2	実装評価	27
4.3	実測用周辺回路	28
4.4	ZyboZ7-20 の P_{LUT} , P_{Reg} 取得	31
4.4.1	提案モデル A の P_{LUT} , P_{Reg} の取得	38
4.4.2	提案モデル B の P_{LUT} , P_{Reg} の取得	39
4.5	DDP 搭載 ZyboZ7-20 電力推定モデルの評価	40
4.5.1	FIR-DDP プログラム	40
4.5.2	検証用 DDP 実装・電力実測	42
4.5.3	提案提案モデル A と提案電力推モデル B を用いて推定電力を導出	43
4.6	考察	47
4.7	結言	48
第 5 章	結論	50
	謝辞	52

目次

参考文献 53

付録 A 55

目次

2.1	STP の基本構成の説明	5
2.2	PS 先行配置型 DDP の構成イメージ	6
2.3	DDP へ入力するパケットの内訳例	6
2.4	Zynq7000 シリーズの FPGA 構成	9
3.1	$T_f + T_r$ の説明	18
3.2	パケット到着時間の違いによる動作頻度の比較イメージ	19
3.3	プログラムの違いによる動作頻度変化の例	19
4.1	FPGA ボードの電力実測環境	28
4.2	測定対象 STP 回路に接続する実測用周辺回路	30
4.3	テスト用 STP の構成	31
4.4	LOC・BEL による配置固定のイメージ	33
4.5	テスト用 STP 回路の PBlock エリア分け	33
4.6	テスト用 STP 回路 DL100 段, インバータ 12 段の配置レイアウト	35
4.7	テスト用 STP 回路 DL50 段, インバータの配置レイアウト	35
4.8	実測電力とモデル A の解析結果のグラフ	36
4.9	FIR-DDP プログラム	41
4.10	16bitDDP の PBlock レイアウト	42

表目次

2.1	PS 先行配置型 DDP の各フィールドの役割	7
2.2	AMD 社の FPGA デバイス Zynq シリーズの主要な素子	10
3.1	AMD 社の FPGA デバイス Zynq シリーズの主要な素子	16
3.2	DDP ステージの各パラメータ取得タイミング	24
4.1	テスト用 STP の各回路パターンと実測電力	36
4.2	提案モデル A に基づく 2 変数重回帰分析の回帰統計	38
4.3	提案モデル A の切片と説明変数 $X_1(N_{LUT})$, $X_2(N_{Reg})$ の回帰係数	38
4.4	提案モデル B に基づく単回帰分析の回帰統計	39
4.5	提案モデル B の切片と説明変数 X_1 の回帰係数	39
4.6	16bitDDP と 32bitDDP の実測電力	43
4.7	16bitDDP の各パラメータと各モデルの推定電力	45
4.8	32bitDDP の各パラメータ	46
4.9	2 つのモデルによる推定値と実測値の比較	47
A.1	テスト用 STP の各回路パターンの実測電力 (1/2)	55
A.2	テスト用 STP の各回路パターンの実測電力 (2/2)	56

第 1 章

序論

1.1 研究背景

近年、IoT の技術進歩により様々な分野に IoT が浸透している [1]。さらには、データをクラウドで処理するのではなく、ネットワーク末端のエッジデバイスでデータ処理を行うエッジコンピューティングも広まりを見せており、市場規模も増加傾向にある [2]。エッジコンピューティングの適用分野はスマートファクトリーやロボットの遠隔操作、高精細映像伝送など多岐に渡る。そのため、エッジコンピューティングで重要な役割を担う IoT エッジデバイスは様々な応用の可能性を考慮し、高い計算性能、省電力性、設計柔軟性が重要である。

これらの問題に対し、データ駆動型プロセッサ（以下 DDP）の FPGA 実装が有望である。DDP は並列処理性能に優れることから高い計算性能を有し、非同期回路であるため省電力性に優れている。FPGA は回路を自由に書き換えることができるため設計柔軟性に優れている。また、様々なラインナップがあり、通常よりも大規模回路を実装できる FPGA、RAM サイズが大きい FPGA、高速動作が可能な FPGA、特定 IP に対応した FPGA など、用途に合わせたデバイスを選択できる。

FPGA 設計にはベンダーが提供する商用 FPGA 設計ツールが主に用いられる。商用 FPGA 設計ツールは、回路設計者が作成した RTL と制約ファイルを基に論理合成や配置配線を半自動的に実行する。また、配線遅延の計算、論理シミュレーション、実遅延シミュレーションが可能で、設計したデバイスの動作・性能確認ができる。このほかに、電力評価機能も有しており、設計した FPGA デバイス電力性能評価もできる。

非同期回路に基づく DDP を搭載した FPGA ボードを電力評価する場合、商用 FPGA 設

1.1 研究背景

計ツールは同期回路に特化しているため電力評価機能の精度が低い。加えて、周辺回路を含めたボード全体の電力を評価できない。そのため、DDP 搭載 FPGA ボードの電力評価には、ワットチェッカーや電力計等を用いた FPGA ボード全体の電力を実測する手法を使用するのが現実的である。

一般的に、デバイスの電力は仕様設計の段階で基準が定められたのち、FPGA 設計工程の要所々々で電力評価を行い、基準を超えた場合には必要に応じて回路構成の変更や機能の見直しが行われる。FPGA ボードの電力を実測するには、FPGA 回路の構成を決定し、RTL を作成後、論理合成、論理シミュレーション、配置配線、実遅延シミュレーション、実機実装・実機テストの順に FPGA 設計工程の全てを終える必要がある。そのため、FPGA 設計工程の途中で電力を評価したい場合は、実測による電力評価では対応できない。

商用 FPGA 設計ツールを用いた同期回路の設計工程では、論理合成、配置配線、実遅延配置配線シミュレーションの各工程で電力推定機能を使用した電力評価が可能である。推定精度は設計工程後期になるほど高くなる。論理合成、配置配線後の電力推定は、実遅延配置配線シミュレーションに比べると精度は劣るものの、FPGA 設計工程のより早期に回路の大まかな電力を推定できるメリットがある。非同期回路の場合はこれに代わる実用的な方法が見当たらない。

FPGA の電力を推定する方法は様々な手法が検討されているものの、FPGA の電力推定には FPGA デバイスと FPGA デバイスに実装する FPGA 回路、両者を考慮するため、手法に限りがある。その中でも、回路構成、動作周波数、動作確率、FPGA のデバイス特性など FPGA 回路の電力に影響する主要素を基に FPGA デバイスの電力をモデル化する手法は、モデルを工夫することで非同期回路にも適用でき、FPGA 設計工程の早期から利用できる評価手法である。

本研究では、汎用的な非同期回路の電力推定を目的とせず、STP-DDP に限定することによって、その動作特性の Heuristic 知識を使って推定精度を上げられるのではないかと考えた。

1.2 提案する DDP 搭載 FPGA ボードの電力評価手法

本研究では、独自の DDP 搭載 FPGA ボード電力推定モデルに基づく電力推定手法を検討した。電力推定モデルで使用するパラメータの中には、デバイスの電力特性も含まれ、これはデバイス毎に固有である。そのため本研究では、対象の FPGA ボードを Digilent 社の ZyboZ7-20 に設定し、DDP 搭載 ZyboZ7-20 の電力推定モデルを構築する。

本提案手法により、早期の電力評価が可能になることに加えて、設計工程早期の電力評価は本提案手法を使用し、最終的なデバイスの評価には実測値を用いるなど、段階に応じた電力評価方法の使い分けが可能になる。

本論文では、第 2 章では電力推定モデルの検討に必要な要素の説明を行う。第 3 章では、DDP 搭載 FPGA ボードの電力評価モデルを検討する。第 4 章で DDP 搭載 ZyboZ7 - 20 の電力推定モデルの実装評価を行う。最後に第 5 章では、本論文の提案とまとめを行う。

第 2 章

データ駆動型搭載 FPGA ボードを構成する要素と電力評価方法の検討

2.1 緒言

DDP 搭載 FPGA ボードの電力評価法を検討するにあたり、DDP と FPGA ボードの特性を考慮する必要がある。また、DDP はセルフタイム型パイプライン（以下：STP）により実現しているため、STP の特性についても同様に考慮する必要がある。よって本章では、まずは STP について説明を行い、続けて、DDP の構成と動作、特性について説明を行う。そして、FPGA ボードの説明と特性を説明する。最後に、STP、DDP、FPGA ボードの各特性を考慮した DDP 搭載 FPGA ボードの電力評価方法を検討する。

2.2 セルフタイム型パイプライン STP

STP の基本構成を図 2.1 に示す。STP は 1 つのステージがデータ転送制御回路 C 素子（以下 C 素子）、データを格納するデータラッチ（以下 DL）、データを処理する Logic から構成され、これらを複数段連ねることでパイプライン回路を構成している。STP の基本動作は C 素子からデータ送信要求信号 Send（以下 Send）が後段 C 素子へ送られ、後段 C 素子はデータ転送が可能な状態であればデータ転送許可信号 Ack（以下 Ack）を返す。前段 C 素子は Ack を受信するとデータラッチ解放許可信号 CP（以下 CP）を立ち上げ、DL を開放しデータを格納後、Logic でデータの処理を行う。

2.3 データ駆動型プロセッサ DDP

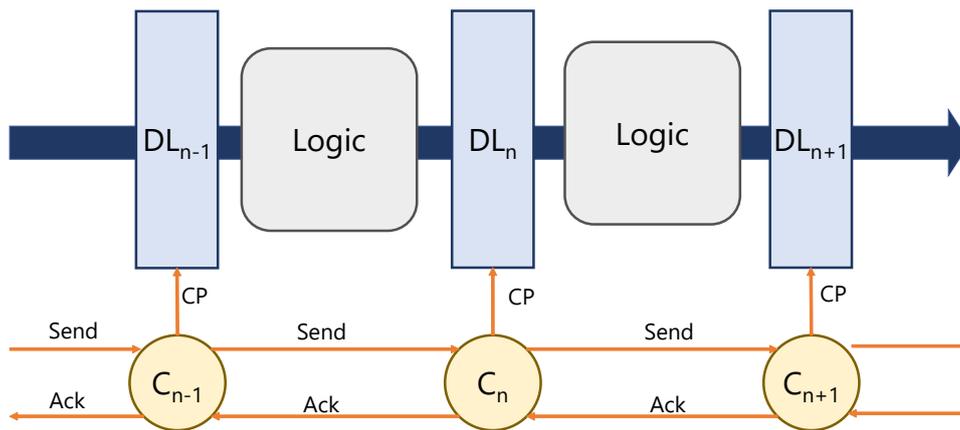


図 2.1 STP の基本構成の説明

このように STP は C 素子がデータ転送を制御してステージが独立に動作する非同期回路である。同期回路では大域クロックが立ち上がると同時に一斉にすべての回路が動作するので積算ピーク電流が大きくなる傾向にあるが、非同期回路は大域クロックが不要であり、各ステージが独立に動作し、回路全体としてバラバラに動作するため積算ピーク電流が小さくなる傾向にあり、省電力である。

2.3 データ駆動型プロセッサ DDP

本研究で使用する PS 先行配置型 DDP[3] の構成図を図 2.2 に示す。そして、DDP 内を周回するパケットのフィールド内訳例を図 2.3 に示し、フィールドの詳細を A.2 に示す。

2.3.1 Mstage

DDP 外部からの入力パケットと DDP 内部を周回するパケットの衝突を防ぐために入力タイミングの調停を行うステージである。CM 素子は C 素子の機能を拡張したもので、DDP 外部と DDP 内部 B ステージの CB 素子と通信を行う。DDP 外部からの Send 信号と DDP 内部 B ステージからの Send 信号の到達タイミングが同時だった場合に、わずかでも早く到達していたパケットを先に受け取り、遅く到達したパケットは待機させることでタイミングの調停を行い、衝突を防ぐ。

2.3 データ駆動型プロセッサ DDP

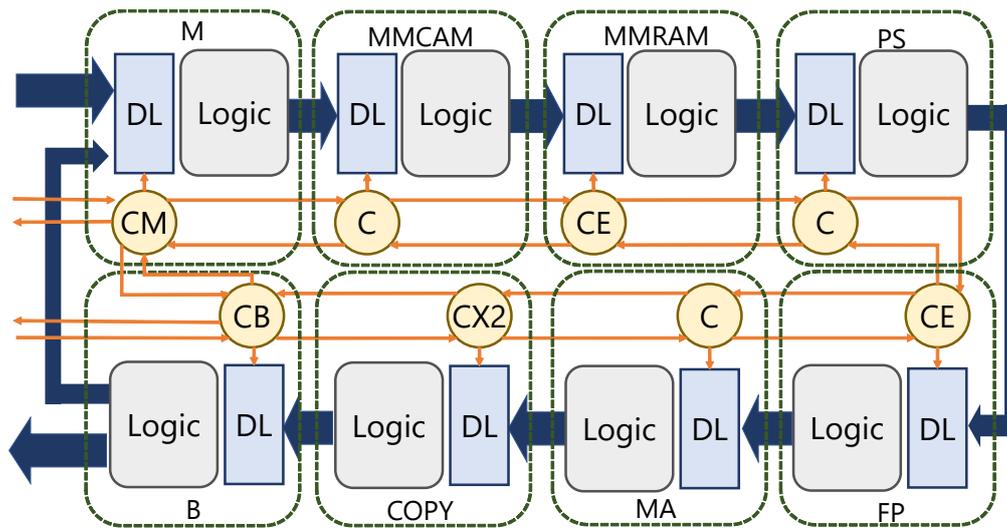


図 2.2 PS 先行配置型 DDP の構成イメージ

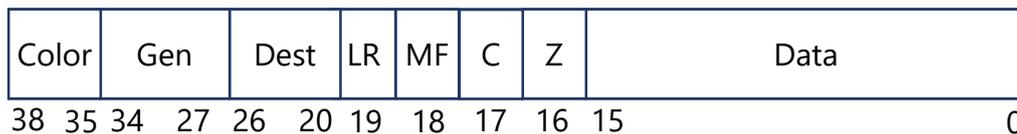


図 2.3 DDP へ入力するパケットの内訳例

2.3.2 MMCAMstage

パケットの待ち合わせ処理を行うステージ。ステージが連想メモリ（CAM:Content Addressable Memory）として動作する。

まず、パケットの MF を確認し、マッチングフラグが立っている場合、CAM の処理へ移行する。マッチングフラグが立っていない場合、そのパケットは待ち合わせが不要なパケットであるため、何も処理せず次のステージへパケットを渡す。

待ち合わせが必要なパケットはパケットの Color, Gen, Dest, LR をアドレスとして CAM にアクセスする。CAM の中に待ち合わせ対象のパケットがある場合：次のステージで対象の Data を受け取る。CAM の中に待ち合わせ対象のパケットが無い場合：Color, Gen, Dest, LR を CAM に記録し、次のステージへ進み Data を RAM へ格納する。

2.3 データ駆動型プロセッサ DDP

表 2.1 PS 先行配置型 DDP の各フィールドの役割

	bit 幅	詳細
Color	3bit	パケットの識別
Gen	8bit	パケットの世代を示す
Dest	7bit	ノードアドレス
LR	1bit	2項演算でそのパケットが左右どちらに属するかを示す
MF	1bit	マッチングフラグ, そのパケットにマッチングが必要かを示す
C	1it	キャリーフラグ, 演算の結果, 桁上げが発生したことを示す
Z	1bit	ゼロフラグ, 分岐判定に用いる
Data	16 bit	そのパケットのデータや, データメモリアクセス時のアドレス
B	1bit	周回フラグ, パケットの宛先が DDP 内部か外部へ出力するかを示す
Copy	1bit	コピーフラグ, パケットのコピーが必要かを示す

2.3.3 MMRAMstage

定数メモリと待ち合わせパケットを格納するためのデータメモリを持つステージである。

パケットの MF が 0 の時, パケットの Dest をアドレスとして, 定数メモリから定数 Data をパケットに結合する。

前ステージで CAM の中に待ち合わせ対象のパケットがあった場合: データメモリから対象の Data をパケットに結合する。前ステージで CAM の中に待ち合わせ対象のパケットが無かった場合: データメモリへデータを書き込む。CE 素子は C 素子の機能を拡張したもので, パケットの削除を行うことができる。前ステージで CAM の中に待ち合わせ対象のパケットが無かったパケットはデータメモリへ書き込んだのち, パケットを削除する。

2.3.4 PSstage

パケットが次の FP ステージ内の ALU で演算が行えるようにパケットのヘッダを書き換えるステージである。プログラムが格納された命令メモリを有しており, パケットの Dest をアドレスとしてメモリへアクセスし, ヘッダを取り出してパケットのヘッダを一部書き換

2.3 データ駆動型プロセッサ DDP

える。書き換えるヘッダにはオペコードや次ノードの Dest, MF, LR, B フラグ, Copy フラグが含まれる。

2.3.5 FPstage

パケットのデータの演算を実行するステージである。演算はオペコードを基に行う。演算は四則演算, シフト演算, 論理演算, 分岐命令, パケット削除, データメモリへの書込みと読み出しがある。分岐命令だった場合は, パケットの C フラグや Z フラグを基に分岐判定をする。パケット削除命令だった場合は, CE 素子がパケットを削除する。

2.3.6 MAstage

データメモリへの書き込み読み出しを行うステージである。前ステージ FP でデータメモリへの書き込み読み出し命令だった場合, このステージのデータメモリに読み書きを行う。それ以外の命令だった場合, 何もせず次のステージへパケットを渡す。

2.3.7 COPYstage

パケットをコピーするステージである。パケットのコピーフラグが立っている場合, パケットをコピーする。コピー後のパケットは Dest を+1 する。CX2 素子は機能を拡張した C 素子である。コピーフラグが立っている場合に前段 C 素子との通信を止め, 後段 C 素子に 2 度転送要求信号 Send を送信する。この時, COPY ステージ内 DL に格納されているパケットは変わっていないため, 2 度同じパケットが送信 = コピーされる。

2.3.8 Bstage

パケットが DDP 内部を周回するか DDP 外部へ出力するかを制御するステージ。CB 素子は C 素子の機能を拡張した素子である。B フラグが立っている場合, M ステージに Send を送信する。B フラグが立っていない場合は, 外部へ転送する。

2.4 FPGA ボード

FPGA (Field Programmable Gate Array) は、多数の書き換え可能な論理ゲートで構成された設計柔軟性の高い半導体デバイスである。回路設計者は、論理ゲートを書き換えることで FPGA 上に任意の回路を実装することができる。近年では、ARM プロセッサと FPGA を組み合わせた SoC が開発・販売されており、より便利な IoT デバイスとして利用されている。この FPGA を搭載し、周辺モジュールを取り付けたのが FPGA ボード、または FPGA 評価ボードである。

FPGA のアーキテクチャは商用 FPGA のベンダ各社で異なるため、ここでは AMD 社の FPGA デバイス Zynq7000 シリーズ [4] を例に挙げ説明する。

Zynq7000 シリーズの FPGA の構成を図 2.4 に示す。

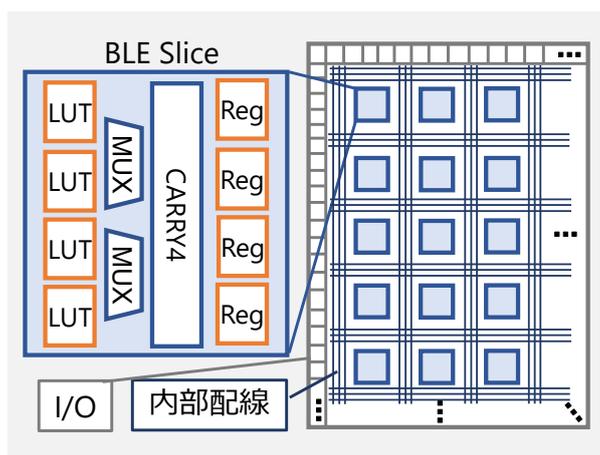


図 2.4 Zynq7000 シリーズの FPGA 構成

Zynq7000 シリーズは BLE スライス構造からなっている。BLE スライスとは、Zynq におけるロジックの集合単位を指す。BLE スライス内部には、LUT (Look Up Table) と Register が多数並んでおり、そのほかに MUX や Carry4 がある。BLE スライス構造はこのスライスが格子状に並べられ、その間を内部配線が通っている。この内部配線は FPGA に回路を実装する際に、BLE スライス間を結ぶ。これにより、FPGA 各所に信号が伝達される。

ベンダの提供する商用 FPGA 設計ツールは、回路設計者が作成した RTL を基に、デジ

2.4 FPGA ボード

タル回路を LUT と Register に置き換えていき、スライス間の配線を自動的にを行い、回路設計者の FPGA 回路設計を支援する。

回路実装時に使用する主要な素子とその機能を表 2.2 に示す。LUT テーブルは複数の入力に対して 1 つの出力を行う素子で、あらかじめ定められた真理値表に従って出力を決定する。Zynq7000 シリーズの LUT は 6 入力 1 出力であるが、必要に応じて FPGA 実装ツールが判断して、2 入力 1 出力、3 入力 1 出力、4 入力 1 出力、5 入力 1 出力といったように、柔軟にこの LUT を使用して回路設計者が定義した回路を FPGA 回路へ合成する。Register は、フリップフロップのことであり、トリガ信号に合わせて信号を格納する。Register は、データラッチとして使用したり、分散メモリとして使用する。Block RAM はデータ保持に特化したメモリモジュールである。プロセッサ実装時は、メモリモジュールとして使用する。

表 2.2 AMD 社の FPGA デバイス Zynq シリーズの主要な素子

モジュール名	機能
LUT	真理値表に従って出力を決定する。Zynq シリーズで 6 入力 1 出力の素子が採用。
Regsiter	フリップフロップのこと。分散メモリとしても使用可能。
BlockRAM	データ保持に特化したメモリモジュール。
DSP	複素乗算器及び浮動小数点乗加算器として利用可能なモジュール。

FPGA ボードの周辺モジュールには、DRAM、スイッチ、ボタンなどの入力端子、無線送受信モジュール、LAN ポート、VGA などの外部出力端子など様々なモジュールが含まれる。これらのモジュールは、FPGA に PIN 接続されており、制約ファイルで PIN を指定することで FPGA 回路と接続して活用することができる。

FPGA ボードの電力には、FPGA が消費する電力に加えて、周辺モジュールが消費する電力も含まれる。そのため、FPGA ボードを IoT エッジデバイスとして用いるにあたり、その電力評価には、FPGA 回路だけでなく周辺モジュールの電力を含めた FPGA ボード全体の電力の評価が必要である。

2.5 DDP 搭載 FPGA ボードの電力評価方法の検討

一般的に、デバイスの電力を評価する手法は電力を実測する方法と、推定する方法に大別される。本セクションでは実測と推定の両手法について説明する。また、DDP 搭載 FPGA ボードの電力評価に用いることを想定したときの、各手法の長短を説明する。

2.5.1 デバイス電力の実測方法

電力を実測する手法は、デバイスの電力をオシロスコープなどで直接計測し、調査するのが一般的である。デバイス電力の実測による電力評価は電力評価方法の中で最も精度が高いため、デバイスの最終的な電力評価に用いられる。

商用 FPGA でオシロスコープを使用する場合は実測ができるように FPGA ボードに手を加える必要があり、コストがかかる。また、デバイスの電力特性はデバイス固有であるため、電力を評価したいすべてのデバイスに手を加える必要があり、現実的でない。

ほかの実測方法として、USB ワットチェッカーがある。これは電源となる USB ポートに接続し、USB ワットチェッカーと FPGA ボードを接続することで FPGA ボードの電力を調べることができる。オシロスコープと違い、FPGA ボードに手を加える必要がないため、ローコストである。欠点は、数 10 ミリ W～数 10W の電力しか調べられないため、精度に難がある。文献 [5] では、DDP の MMCAM の構成を変更し、変更前と変更後 DDP の電力評価に FPGA ボードの電力を USB ワットチェッカーを用いて実測評価を行った。

また、実測手法全体の問題として、実測による評価ができるタイミングは FPGA 設計工程（RTL 設計→論理合成→論理シミュレーション→配置配線→実遅延シミュレーション→実機実装テスト）中、最後の工程である実機実装テストのタイミングである。そのため、計測結果は確実であるが、電力評価のタイミングとしては遅いというのが電力実測の立ち位置である。より早期の電力評価にはデバイスの電力を推定する方法を用いる必要がある。

2.5 DDP 搭載 FPGA ボードの電力評価方法の検討

2.5.2 デバイス電力の推定方法

デバイスの電力を推定する方法は、これまでに様々なアプローチが提案されている [6]。中でも FPGA で用いられているのは、シミュレーションベースの方法、回路構成ベースの手法であり、これは商用 FPGA 設計支援ツール内にも使用されている。

シミュレーションベースの手法は、回路動作をシミュレーションし、これを基に電力を推定する方法である。

回路構成ベースの手法は FPGA へ実装する回路構成をもとに、素子の使用量など考慮して電力を推定する方法である。

AMD 社の Vivado など、ベンダが提供する商用 FPGA 設計ツール内に含まれている電力評価機能は、これら複数の推定方法を組み合わせてより高精度な推定を行っていると考えられる [7]。また、AMD 社はより早期の電力推定目的のスプレッドシートベースの電力推定ツールも公開している [8]。

そのほかの方法として、シミュレーションに基づく方法では、FPGA 向きスパイスシミュレータである FPGA-SPICE が提案されている [9]。

FPGA 上に実装した回路の電力推定の特徴として、より高精度な電力推定には FPGA の半導体プロセスパラメタと FPGA に実装する回路の構成・動作を考慮しなくてはならないことが挙げられる。そのため商用 FPGA は半導体プロセスパラメタが非公開であるため、フリーのツールを用いることは難しく、商用 FPGA を販売するベンダーが提供する商用 FPGA 設計ツールを使用するしかない。

しかし、商用 FPGA 設計ツールはその多くが同期回路向けに特化しており、推定精度が低い。また、FPGA ボードを IoT エッジデバイスとして扱うにあたっては、FPGA ボード全体の電力を評価することが必要だが、商用 FPGA 設計ツールの電力評価機能は FPGA ボード全体の電力を推定することができない問題もある。

デバイス特性が分からない状況に対する電力推定手法のアプローチとして、電力モデリングベースの手法がある。[6]。この手法は、特定パラメータ（動作確率、クロック周波数、

2.6 結言

デバイス特性、回路動作など)を使用し、電力モデルを作成する。そして、作成した電力モデルを用いて、電力を推定する。この時、デバイス特性や回路動作など、不明なデータはシミュレーションや測定からデータを取得する。

Gaurav Verma らは、低電力で動作する FPGA 回路は動作が複雑な場合があり、ベンダが提供する電力推定ツールでは実装した回路の電力が複雑な場合うまく推定できない可能性があることを背景に、低電力で動作する同期型 FPGA 回路を対象とした独自の FPGA 電力モデルを提案している [10]。この提案モデルでは、Xilinx ISE で使用可能な IP コアを対象に精度検証を行った結果、平均百分率誤差 3.73%の精度を報告している。

DDP 搭載 FPGA ボードの電力推定モデルの研究の一環として、岡村が行った研究では、FPGA 実装した STP 回路の簡易版電力推定モデルを提案している [11]。このモデルは、STP 回路を実装した FPGA の電力を見積るために、STP の DL 段数と動作頻度に着目しており、パラメタとして、パイプライン段数、パケット長、C素子の動作頻度を掛けた値を用いて、STP を実装した FPGA の電力を推定している。しかし、この方法を DDP に適用するには、DDP の Logic の回路構成や回路動作の電力への影響を考慮できていないこと、電力推定の対象が FPGA に限られていることなど問題がある。

そのため、本研究では、独自の DDP 搭載 FPGA ボードの電力推定モデルを作成し、そのモデルに基づいて電力推定を行う手法を検討した。

2.6 結言

本章では、まずは STP について説明を行い、DDP の構成と特性、各ステージの動作について説明を行った。そして、FPGA ボードの説明として Zynq7000 シリーズを例に挙げその概要と特性を説明した。最後に、STP、DDP、FPGA ボードの各特性を考慮した DDP 搭載 FPGA ボードの電力評価方法を検討した。次章では、独自の DDP 搭載 FPGA ボードの電力推定モデルを検討し、そのモデルに基づいて電力推定を行う手法を提案する。

本研究では、DDP 搭載 FPGA ボードの電力推定モデルに基づく電力推定法を提案した

2.6 結言

が、他の手法の検討や、本提案手法と他の手法の比較検討などが残されている。

第 3 章

DDP 搭載 FPGA ボードの電力推定法

3.1 緒言

本章では、初めに、電力に影響する主な要素である回路構成、動作周波数、動作確率、FPGA デバイスの電力特性について、DDP を FPGA に実装した際の影響について述べる。そして、それを踏まえて DDP 搭載 FPGA ボードの電力推定モデルを検討し、そのモデルを用いて電力推定を行う方法を提案する。

3.2 DDP 搭載 FPGA ボードの電力推定モデルの検討

DDP を実装した FPGA 回路の電力をモデル化するにあたって電力へ影響する主な要素を検討した。一般的に、回路構成、動作周波数、動作確率、FPGA デバイスの電力特性が電力へ影響する要素として挙げられる。

FPGA に実装する DDP は 2.3 章で述べたように、機能の異なる複数のステージで構成されている。そのため、ステージ毎に回路構成、パケットの処理、単位時間当たりの動作頻度が異なる。また、DDP は非同期パイプライン回路である STP により実現されているので、ステージ毎に動作タイミングが異なる。そのため、DDP 搭載 FPGA ボードの電力推定モデルでは DDP のステージ毎の電力を積算する必要がある。

3.2 DDP 搭載 FPGA ボードの電力推定モデルの検討

3.2.1 回路構成

前述したように、DDP の各ステージは主に DL, Logic, C 素子から構成される。FPGA に DDP を実装する際、DDP の DL はそのすべてが Register により構成され、Logic は LUT により構成され、C 素子は LUT とごく少数の Register により構成される。

AMD 社の Zynq7000 シリーズに Data 部が 16Bbit の DDP を実装した際の FPGA リソース使用量を表 3.1 に示す。この表は、各ステージのロジック・DL のリソース使用量と、各ステージの C 素子又は拡張 C 素子のリソース使用量を分けて示している。

表 3.1 AMD 社の FPGA デバイス Zynq シリーズの主要な素子

	LUT	Register	Block RAM	DSP
最大素子数 (Zynq7020)	53,200	106,400	140	220
DDP で使用する素子の数	1,987	1,722	2.5	1
M	28	76	0	0
(CM)	92	0	0	0
MMCAM	745	1,326	0	0
(C)	35	0	0	0
MMRAM	95	38	1	0
(CE)	35	1	0	0
PS	1	52	0.5	0
(C)	35	0	0	0
FP	658	62	0	1
(CE)	28	1	0	0
MA	61	41	0.5	0
(C)	44	0	0	0
COPY	15	40	0	0
(CX2)	60	8	0	0
B	0	38	0.5	0
(CB)	34	1	0	0

各ステージの構成のうち、MMCAM は例外的に Logic 内で CAM を構成するのに

3.2 DDP 搭載 FPGA ボードの電力推定モデルの検討

Register を分散メモリとして使用しているため、他のステージに比べて Register の数が多くなっている。

また、MMRAM, PS, MA, B ステージ内のメモリは Zynq7000 シリーズでは Block RAM で構成され、FP ステージ内の ALU 内の乗算器は DSP により構成される。

表より、各ステージ内での LUT や Register の数は、Block RAM や DSP に比べて多いことが分かる。よって、DDP の各ステージの構成素子は LUT と Register が支配的であり、主要な電力推定パラメータであると考えられる。

以上のことから FPGA に DDP を実装する際の回路構成は主に LUT と Register で構成されており、そのステージの LUT 数 N_{LUT} と Register 数 N_{Reg} が各ステージの電力を決定する主要な電力推定パラメータであると考えられる。

特殊な例として、100 個の BlockRAM からなる大規模メモリを持つような DDP アーキテクチャや、複雑な計算を行うため ALU で 200 個の DSP を使用するような特定用途に特化した DDP アーキテクチャを対象に含む電力推定モデルを検討する場合は、Block RAM や DSP も LUT や Register に並ぶ電力推定パラメータとして考える必要がある。

しかし、前述したように電力モデル化にあたってはデバイス特性など不明なパラメータがある場合は、事前のシミュレーションや測定が必要であり、パラメータの増加は、事前のシミュレーションや測定作業等の作業量増加に直結する。ベーシックな DDP アーキテクチャでは、Block RAM や DSP の電力への影響は小さいと考えられるなか、特殊な例に対応するために作業量を増やすのは非効率的であり、望ましくない。そこで、本研究では、まずは、特殊な例を考慮せず、ベーシックな DDP アーキテクチャを電力推定モデルの対象として電力推定モデルを検討する。

3.2.2 動作周波数

同期回路においては、大域クロックがその回路の動作トリガとなるので、大域クロックの周波数がその回路の動作周波数である。

対して、非同期回路である DDP には動作周波数はないが、電力推定モデルの検討にあ

3.2 DDP 搭載 FPGA ボードの電力推定モデルの検討

たって各ステージの動作頻度を示すパラメータが必要である。そこで、疑似的な動作周波数 f を導入し、これを、 $f \doteq 1/(\text{周期})$ とする。

周期には、回路に依存する周期、アプリケーションの実行周期、DDP プログラムの構造の3つの要因が関係しており、各要因に分けて考える必要がある。

回路に依存する周期は、C素子のパケット転送周期 $T_f + T_r$ により決定する。図 3.1 は、隣接する C 素子 C_0, C_1, C_2 が続けて入力された2つのパケットを後段 C 素子へ転送する際の信号伝達の流れを示している。主な流れは、初めに、C 素子 C_0 の Send が C_1 に送信する。次に、 C_1 はパケットを受け入れ可能な状態なら C_0 に向かって Ack を返す。最後に、 C_0 は Ack を受けたのち、CP を立ち上げる。 C_1 は同様の手順で C_2 へパケットを転送する。 C_1 が C_2 へパケット転送後、 C_0 は再び C_1 へパケットが転送可能になる。このように、パケットが立て続けに入力された状況下で、パケットを転送するのに時間を T_f 、そこから次のパケットを転送するのに必要な時間を T_r としたとき、これらを合わせた値が $T_f + T_r$ である。

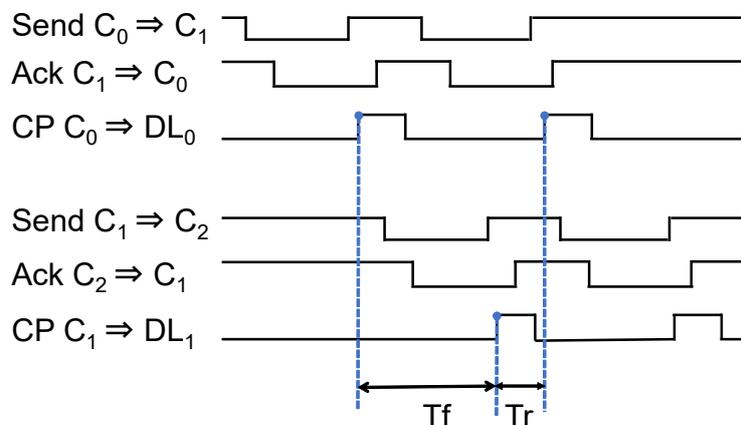


図 3.1 $T_f + T_r$ の説明

$T_f + T_r$ は、C 素子の信号タイミングの設計時に調節される。

到着したパケットの処理にかかる時間はステージ毎に異なるが、C 素子は前段ステージから Send が来ても、受け入れ可能でなければ Ack を返さないため、最も処理が遅いステージに動作時間が律速されるという特徴がある。本研究では $T_f + T_r$ を 50ns とした。

次に、DDP を用いたアプリケーションの実態は、パケットが入力された DDP は、一連

3.2 DDP 搭載 FPGA ボードの電力推定モデルの検討

の DDP プログラムが実行され、パケットが出力されるか待ち合わせ状態になると DDP は動作しなくなり、休んだ状態になる。これをアプリケーションの 1 周期としたとき、アプリケーションの実行周期はパケット入力周期にも依存すると言える。

このようなアプリケーション実行周期の違いによる f への影響のイメージを図 3.2 に示す。パケット入力周期が異なるアプリケーション A とアプリケーション B があり、アプリケーション A のパケット入力周期がアプリケーション B の 2 倍とする。この場合、アプリケーション A の DDP 各ステージはアプリケーション B の DDP 各ステージに比べて動作頻度が 2 倍となるため、 f も 2 倍となる。

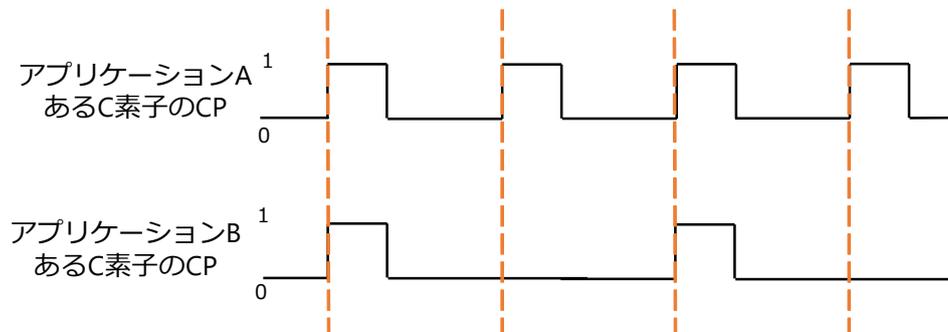


図 3.2 パケット到着時間の違いによる動作頻度の比較イメージ

最後に、DDP プログラムの構造による f の違いのイメージを図 3.3 に示す。この図は、2 つのパケットが待ち合わせをして演算を行う時の MMRAM ステージと PS ステージの転送許可信号 CP が立ち上がる回数の差を示している。

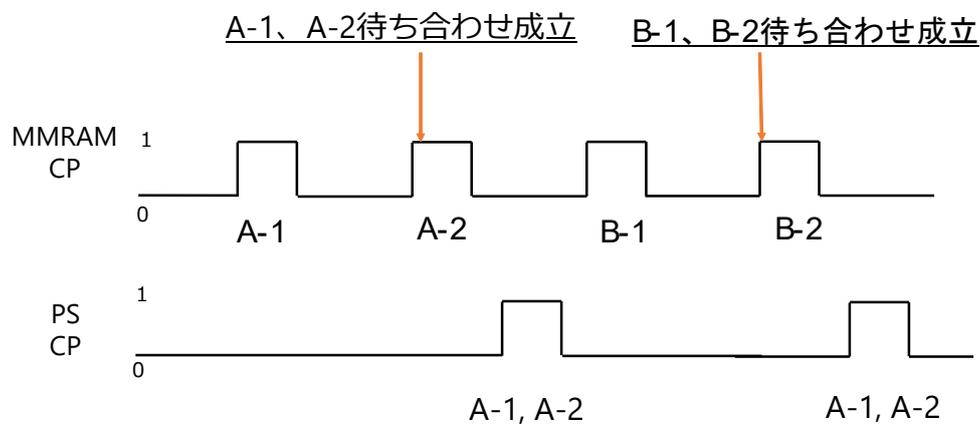


図 3.3 プログラムの違いによる動作頻度変化の例

3.2 DDP 搭載 FPGA ボードの電力推定モデルの検討

このような演算では、マッチング処理が行われる MMRAM とそれ以前のステージと、マッチング処理が行われた後の MMRAM 以降のデータの流量が異なる。つまり、M, MMCAM, MMRAM ステージのデータの流量を X としたときに、PS ステージ以降の流量は $X/2$ となる。よって、待ち合わせ処理がある DDP プログラムの特性として、単位時間当たりの動作頻度が、M, MMCAM, MMRAM ステージでは、(マッチングが不要なデータ) + (マッチングが必要なデータ)[回] に対して、PS 以降のステージは (マッチングが不要なデータ) + (マッチングが必要なデータ)/2[回] となる。

また、COPY ステージでコピーをする場合、COPY ステージ以前のデータの流量が X の時、COPY ステージとそれ以降のステージはデータ流量が $2X$ となる。そのため、単位時間当たりの動作頻度は COPY ステージ以前のステージでは (コピーしないデータ) + (コピーするデータ)[回] なのに対して、COPY ステージとそれ以降のステージは (コピーしないデータ) + (コピーするデータ) \times 2[回] となる。

このように、周期には、回路に依存する周期とアプリケーションの実行周期、DDP プログラムの構造が影響するため、これらを加味して疑似的な動作周波数 f を決定する。

3.2.3 動作確率

ステージにデータが到着したときの Logic と DL の動作確率、つまり、LUT, Reg の動作確率は、DDP のアーキテクチャと DDP プログラム構造により決定する。このとき、LUT と Register の動作確率はステージ毎に異なる。また、同一ステージ内の LUT と Register の動作確率も異なる。よって、LUT の動作確率を α_{LUT} 、Register の動作確率を α_{Reg} とし、ステージ毎にこれらの動作確率を検討する。

α_{LUT} , α_{Reg} は DDP アーキテクチャと DDP プログラムの構造により変化するため、ここでは、本研究で利用した PS 先行配置型 DDP の各ステージの α_{LUT} , α_{Reg} について検討した。

まず、各ステージの転送制御を担う C 素子及び拡張 C 素子は、1つのデータに対して C 素子が 1 度動作するため、動作確率は 1 であるとする。C 素子は LUT で構成されるの

3.2 DDP 搭載 FPGA ボードの電力推定モデルの検討

で、C 素子を構成する LUT の動作確率 α_{LUT} は 1 である。

通常、各ステージの DL は格納している N bit パケットに対して、すべての bit が反転したパケットが転送されてくると DL の動作確率は 1 となり、すべての bit が同じパケットが転送された場合、DL で格納しているパケットが変化しないため、動作確率は 0 となる。この時、bit が反転している割合により各ステージの DL の動作確率 α_{Reg} が決定し、これは 0~1 になる。

次に、各ステージの logic つまり LUT や M ステージ、MMCAM ステージ等の特殊な DL の動作確率について検討した。

M ステージの DL は、内部周回パケットを受ける DL と入力パケットを受ける DL からなる。CM 素子が動作する際、どちらか一方の DL のみ動作するため、 α_{Reg} は最大で 1/2 である。これは、すべての bit が反転している時に限った話であり、実際は、bit が反転している割合によって決まり、0~1/2 になる。また、Logic は、内部周回パケット、入力パケットに関係なく動作するが、DL と同様に bit が反転している割合により決まり、 α_{LUT} は 0~1 になる。

MMCAM は、パケットを格納する DL のほかに、待ち合わせパケットのヘッダ情報を格納する DL が Logic 内にある。本研究で使用している DDP ではこのヘッダを格納する DL が 64 個ある。この数は可変であるため、DDP アーキテクチャ次第で変化する。この DL はパケットのマッチングフラグ MF が 1 の時にのみ動作し、ヘッダ情報を格納するため、パケットが待ち合わせを必要としない限りは動作しない。そのため、待ち合わせを必要としない DDP プログラムでは、MMCAM の Logic 内の α_{Reg} は 0 である。対して、パケットの待ち合わせが必要でパケットのヘッダを格納する場合は、64 個の DL のうちいずれか 1 か所の DL のみが動作するため、 α_{Reg} は 1/64 である。 α_{Reg} は、DDP プログラム中で待ち合わせ処理が行われる割合に依存しており、DDP プログラム構造構造により決定する。Logic には、ヘッダ情報の比較機構やヘッダを格納する DL を管理している VALID、パケットの待ち合わせ成立時の発火処理を行う FIRE などがある。ヘッダ情報の比較機構は、新たなパケットがステージに転送される度に動作するが、それ以外のモジュールはいずれもパケット

3.2 DDP 搭載 FPGA ボードの電力推定モデルの検討

の待ち合わせがある場合は動作し、そうでないときは動作しない。 α_{LUT} は、DDP プログラム中で待ち合わせ処理が行われる割合に依存しており、DDP プログラム構造により決定する。

MMRAM ステージの Logic は、パケットが転送後ヘッダのアドレスを基に、待ち合わせパケットのデータを格納しているメモリと、定数演算用の定数メモリへアクセスする。そして、パケットの MF により、どちらのメモリからの出力データをパケットに接合するかを判断する。このため、プログラム構造の影響を受けず、MF が 0, 1 どちらであっても両メモリからデータを出しパケットへ接合するため、新たなパケットが入力される度に各 Logic は動作する。よって、 α_{LUT} はパケット、両メモリからの出力 bit が反転している割合によりきまり、これは 0~1 となる。

PS ステージは、プログラム構造の影響を受けず、パケットが到着するとヘッダの dest をアドレスとして命令メモリにアクセスし、命令メモリから出力された新たなヘッダ情報でパケットの一部を書き換えるという一連の動作を行うため、 α_{LUT} は、プログラム構造によらず、bit が反転している割合により決まり、これは 0~1 となる。

FP ステージは、ステージ内の ALU の設計方針が、全ての演算後オペコードで演算結果を選択出力するのか、もしくは、オペコードで必要な演算のみを選択し実行するのかで α_{LUT} が変化する。つまり、 α_{LUT} は、DDP アーキテクチャとプログラム構造と、bit が反転している割合により決まり、これは、0~1 となる。

MA ステージは、データメモリを有しているが、前段ステージ FP からデータメモリへの書込み/読み出し命令が送られてこない限り logic が動作しない。この場合の α_{LUT} は 0 である。書込み/読み出し命令時には、転送されてきたパケット、メモリから読み出したデータの bit の反転割合により決定し、 α_{LUT} は最大で 1 となる。よって、 α_{LUT} は、書込み/読み出し命令と bit の反転割合により決定し、これはプログラム構造の影響を受ける。

COPY ステージでは、パケット複製時、DL は動作せず、Logic は動作する。複製しない時は、DL, Logic 共に動作し、その動作確率は bit の反転の割合により決定する。つまり、プログラム構造に依存する。よって、 α_{LUT} と α_{Reg} は、DDP プログラム中のパケット複製

3.2 DDP 搭載 FPGA ボードの電力推定モデルの検討

命令の割合と bit の反転割合により決定し、これは 0~1 となる。

B ステージは、ステージ内のパケットを外部に出力するか、内部周回させるかの分流については CB 素子が行うため、分流によるロジック及び DL の影響はない。したがって、B ステージはプログラム構造の影響を受けず、 α_{LUT} は bit の反転割合により決定し、0~1 となる。

3.2.4 FPGA デバイスの電力特性

FPGA の半導体プロセスパラメータは非公開なため、LUT の電力 P_{LUT} と Register の電力 P_{Reg} は不明である。そのため、電力推定モデル作成のために P_{LUT} , P_{Reg} を取得する必要がある。

そこで、本研究では、TEG の考え方を参考にした TEG 回路を用いてデバイスの P_{LUT} , P_{Reg} を推定する方法を検討した。

TEG は規則的かつ簡単な回路パターンからデバイス特性を調査する手法で、半導体デバイス設計時のテストに用いられている。

この TEG を参考に、本研究では、動作周波数 f , LUT 数 N_{LUT} , Register 数 N_{Reg} をパラメータとして、パラメータパターンが異なる複数のテスト回路 (TEG 回路) の電力を実測し、パラメータと実測電力から P_{LUT} , P_{Reg} を推定する。

TEG 回路には STP 回路を用いる。その理由として、同期回路は大域クロックが立ち上がると一斉に回路が動作するため、電力を波形で考えると、クロックに合わせて急激に波が大きくなる。対して、DDP を実現している STP は各ステージが独立して動作するため、波の上下が比較的が穏やかであり、同じ周波数で動作したときに、同期回路よりも非同期回路である STP のほうが電力が小さくなると考えられる。TEG 回路の実測電力から P_{LUT} , P_{Reg} を類推するにあたって、この特性を考慮するべきであるため、DDP を実現している STP 回路を TEG 回路として用いる。

そのため、TEG 回路は、C 素子、Logic、DL から構成される。回路パターン変更時には、C 素子の $T_f + T_r$ を調節することで f を変化させ、Logic を変更することで LUT 数を変化

3.3 DDP 搭載 FPGA ボードの電力推定法

させ、DL を変更することで Register を変化させる。

実測電力から P_{LUT} , P_{Reg} の推定には、回帰分析による解析を行う。回帰分析には、目的変数と説明変数に実測電力、 f , N_{LUT} , N_{Reg} 用いる。回帰分析により得られた係数を P_{LUT} , P_{Reg} として、切片は周辺回路やボードの電力に関するものとする。

ある FPGA デバイスの P_{LUT} , P_{Reg} と切片は固有であり、実装する回路に依存しない不変的なパラメータと仮定する。したがって、複数のデバイスで電力推定モデルを適用する場合は、デバイス毎に P_{LUT} , P_{Reg} を取得する必要がある。

3.3 DDP 搭載 FPGA ボードの電力推定法

以上のことから本研究では、各ステージの電力を P_{stage} とし、すべてのステージの P_{stage} を積算した値を DDP 全体の電力とし、そこに周辺回路やボード電力、Static 電力からなる P_{others} を含めた値を DDP 搭載 FPGA ボードの電力 $P_{DDPonFPGA}$ と定式化した。定式化した電力推定モデルを式 3.1 に示す。

$$P_{DDPonFPGA} = \sum^{stage} P_{stage} + P_{others} \quad (3.1)$$

$$P_{stage} = f(\alpha_{LUT}P_{LUT}N_{LUT} + \alpha_{Reg}P_{Reg}N_{Reg})$$

DDP 各ステージのパラメータは、LUT・Register の数 $N_{LUT} \cdot N_{Reg}$ は FPGA 実装時（配置配線時）、動作周波数 f は、FPGA 回路設計時、動作確率 $\alpha_{LUT} \cdot \alpha_{Reg}$ は仮動作時、デバイスの電力特性 P_{LUT} , P_{Reg} は事前に TEG 回路を用いた実測時に取得したものを使用する。TEG と DDP 各ステージのパラメータを所得するタイミングを表 3.2 にまとめる。

表 3.2 DDP ステージの各パラメータ取得タイミング

パラメータ	TEG	DDPstage
① N_{LUT} , N_{Reg}	実装時	実装時
② f	設計時	設計時
③ α_{LUT} , α_{Reg}	1	仮動作時
④ P_{LUT} , P_{Reg}	TEG	実測時

3.3 DDP 搭載 FPGA ボードの電力推定法

本モデルは、DDP 搭載 FPGA ボードの基礎的な電力推定モデルである。実際に、電力推定モデルを用いた電力推定にあたりいくつかの解釈が可能である。本研究では2つの解釈として、基礎に従ったモデル A と、導出過程を簡潔にしたモデル B を検討した。

3.3.1 DDP 搭載 FPGA ボード電力推定モデル A

式 3.1 の P_{LUT} , P_{Reg} を TEG 回路の実測データを用いて回帰分析で係数を求めるには、FPGA 回路の動作時電力とそれ以外の電力を切り分ける必要がある。

そのため、TEG 回路であるテスト用 STP 回路の電力を実測する際に、テスト用 STP 回路が動作している時の電力と、テスト用 STP 回路の C 素子の停止させて FPGA 回路が動作していない状態の電力である STP 回路停止時電力を実測しておく。そして、動作時電力を (動作時電力) = ((実測電力) - (STP 停止時電力))/ f とし、この動作時電力を回帰分析の説明変数とする。

そして、説明変数を $N_{LUT}:X_1$, $N_{Reg}:X_2$ として2変数重回帰分析を行う。これにより得られた係数 X_1 , X_2 をそれぞれ P_{LUT} , P_{Reg} とする。また、得られた切片は、STP 回路停止時電力と合わせて P_{others} とする。

これらをまとめた電力推定モデル A (以下：提案モデル A) を式 3.2 に示す。

$$\begin{aligned} P_{DDPonFPGA} &= \sum^{stage} P_{stage} + P_{others} \\ P_{stage} &= f(\alpha_{LUT} X_1 N_{LUT} + \alpha_{Reg} X_2 N_{Reg}) \\ P_{others} &= (\text{切片}) + (STP \text{ 停止時電力}) \end{aligned} \quad (3.2)$$

3.3.2 DDP 搭載 FPGA ボード電力推定モデル B

DDP の各ステージを構成する主要な電力パラメータである LUT と Register は、トランジスタレベルで考えると、どちらも CMOS で構成されているため、素子当たりの電力差は極めて小さいと考えられる。そこで、 $P_{LUT} = P_{Reg}$ とみなすことで、電力推定モデルが簡潔になり、導出過程も簡潔になる。

電力推定モデル B(式 3.3) (以下：提案モデル B) では、目的変数を実測電力に、説明変

3.4 結言

数を $f \times (N_{LUT} + N_{Reg}) : X_1$ とし、単回帰分析を行う。得られた切片を P_{others} とする。 f , N_{LUT} , N_{Reg} をまとめて説明変数とすることで、計算過程で STP 停止時電力が不要になるため、それにより実測回数を減らすことができ、導出過程がより簡潔になっている。

$$\begin{aligned} P_{DDPonFPGA} &= \sum^{stage} P_{stage} + P_{others} \\ P_{stage} &= X_1 f (\alpha_{LUT} N_{LUT} + \alpha_{Reg} N_{Reg}) \\ P_{others} &= (\text{切片}) \end{aligned} \quad (3.3)$$

3.4 結言

本章では、初めに電力に影響する主な要素として、回路構成、動作周波数、動作確率、FPGA デバイスの電力特性を挙げ、DDP を FPGA に実装した際の影響について述べた。そして、それを踏まえて DDP 搭載 FPGA ボードの基本となる電力推定モデルを検討し、基本のモデルを用いて電力推定するにあたり、仮定したモデルがいくつかの解釈ができることを説明した。最後に、解釈の異なる 2 つのモデルとして、基本のモデルに従ったモデル A と、より簡潔なモデルであるモデル B を提案し、これら 2 つのモデル用いて電力推定を行う方法を提案した。次章では、提案モデル A と提案モデル B の実装評価を行う。

本章では、PS 先行配置型 DDP を対象に動作確率を検討したが、これは、異なるアーキテクチャに適用することはできない。そのため、異なる DDP アーキテクチャの動作確率の検討が残されている。

また、 P_{LUT} と P_{Reg} を求めるために、TEG 回路の実測結果を、電力推定モデル A では 2 変数重回帰分析にかけ、電力推定モデル B に単回帰分析かける方法を提案したが、解析方法やデータの前処理により P_{LUT} と P_{Reg} は変動しうる。本研究ではどのような解析手法が最適であるかの議論ができていないため、今後、解析方法について議論が必要である。

第 4 章

実装評価

4.1 緒言

本章では、前章で提案した提案モデル A と提案モデル B の実装評価を行う。まずは、デバイスの P_{LUT} , P_{Reg} を取得するため、TEG 回路を実装用周辺回路と共に FPGA ボードに実装し、電力を測定する。そして、得られた実測電力, f , N_{LUT} , N_{Reg} を基に、提案モデル A と提案モデル B に合わせ回帰分析を行い、デバイスの P_{LUT} , P_{Reg} を取得する。最後に、DDP 搭載 ZyboZ7-20 を対象に提案モデル A と提案モデル B, 実測電力の 3 者間で電力を比較する。

4.2 実装評価

本研究では、Degidlent 社 ZyboZ7-20 を対象に提案モデル A, B の実装評価を行った。ZyboZ7-20 には FPGA として AMD 社の Zynq7020 が搭載されている。テスト用 STP 及び評価用 DDP の実装には、商用 FPGA 設計ツールである AMD 社の Vivado 2022 ML edition を使用した。

実装評価時の ZyboZ7-20 の電力の計測には RouteR 社の USB ワットチェッカーを使用した。この USB ワットチェッカーは、電源と計測対象の間に設置し、計測対象が消費する電力を計測する。また、端末ディスプレイ上に W, A, V, mWh, 抵抗, 温度, 計測時間などのデータが表示される。時間以外のデータは 1 秒間に 2 回更新される。

実測評価の環境を図 4.1 に示す。

4.3 実測用周辺回路

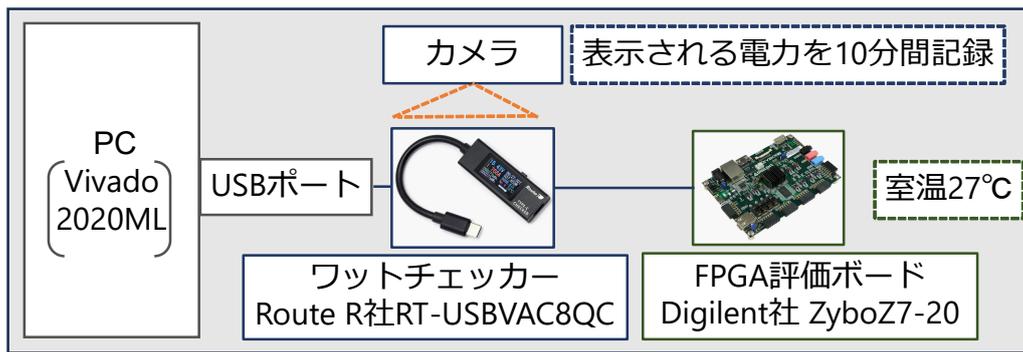


図 4.1 FPGA ボードの電力実測環境

電源である USB ポートに USB ワットチェッカーを接続し、これと FPGA ボードを接続する。ワットチェッカーに表示されるデータはカメラを使用して 10 分間記録した。10 分記録したうち 9 分 31 秒から～10 分 00 秒のデータつまり、60 個のデータの平均をその回路の電力とした。室温は 27～25°C で実験を行った。

実装評価の流れとして、初めに TEG 回路を ZyboZ7-20 に実装し、電力を実測する。その結果を提案モデル A、提案モデル B に合わせて回帰分析を行い、ZyboZ7-20 の P_{LUT} 、 P_{Reg} を取得する。そして、評価用 DDP を ZyboZ7-20 に実装し、電力を実測する。評価用 DDP の実装時に f 、 N_{LUT} 、 N_{Reg} 、 α_{LUT} 、 α_{Reg} を取得または決定し、これらパラメータを提案モデル A と提案モデル B にそれぞれ適用し、電力を推定する。最後に、実測した電力と推定した電力を比較する。

TEG 回路と評価用 DDP を ZyboZ7-20 に実装する際は、動作確認、実測時の条件を揃えるために共通の実測用周辺回路を取り付けて電力を実測した。

4.3 実測用周辺回路

FPGA 回路を FPGA ボードに実装した後、その回路が正常に動作をしているかを確認をする必要がある。Vivado には FPGA ボード実装後にデバッグを行うデバッグツールがあるが、TEG 回路や評価用 DDP は非同期回路であるため、Vivado のデバッグツールを用いたデバッグでは正しくデバッグできていると保証できない。そこで、本研究では、TEG

4.3 実測用周辺回路

回路や評価用 DDP の出力の一部を ZyboZ7-20 の 7seg-LED に出力し、出力された値の正誤判定を目測により行う方法を用いる。この方法は、デバッグ用の追加回路が余分な電力を消費する欠点があるものの、デバッグに必要な回路は、回路規模が非常に小さく計測対象の TEG 回路や評価用 DDP に比べ数%程度であるため、誤差と言える。そのうえ、TEG 回路や評価用 DDP で共通の実測用周辺回路を用いるため、このデバッグ回路が消費する電力は周辺回路の電力として P_{others} に含まれると考えられる。これらのことから、実測用周辺回路の電力は、 P_{LUT} 、 P_{Reg} への影響は極めて小さいと考えられる。

TEG 回路や評価用 DDP の動作確認は、各ステージの Logic でデータが正しく処理されて DL に格納できているか確認が必要である。

加えて、TEG 回路や評価用 DDP を構成する C 素子の $T_f + T_r$ が実遅延実配置配線シミュレーションで確認したときと、実機に実装したときで異なることが分かっている。この現象は、実遅延配置配線シミュレーションと実際の FPGA で $T_f + T_r$ に影響する C 素子間の配線遅延が異なることが原因である。この問題の背景には、実遅延実配置配線シミュレーションで用いられている半導体プロセスデータと実際の FPGA ボードに搭載されている FPGA チップの半導体プロセスデータがわずかに異なるというものがある。この現象は、本研究で使用する ZyboZ7-20 でも発生する。そのため、実遅延実配置配線シミュレーションで C 素子が正常に動作していることを確認しても実際の FPGA 上では正常に動作していない場合がある。

これらを踏まえて、本研究では、実配置配線後の実遅延シミュレーションで $T_f + T_r$ を確認したうえで、カウンタを用いて C 素子の $T_f + T_r$ を計測・出力し、許容範囲に収まっているか確認する。

よって、実測用周辺回路では、データが正しく処理されているかを確認できるよう TEG 回路や評価用 DDP つまり、STP 回路から出力されるデータの一部を 7seg-LED に出力するようにし、さらに、C 素子の $T_f + T_r$ を確認できるようカウンタが出力する時間を 7seg-LED に出力する。7seg-LED は電力消費が大きいいため、動作確認時は 7seg-LED を取り付けて確認に用いて、電力測定時は、7seg-LED は FPGA 回路動作に不要なため、取り

4.3 実測用周辺回路

外した。

実測で用いた実測用回路を図 4.2 に示す。

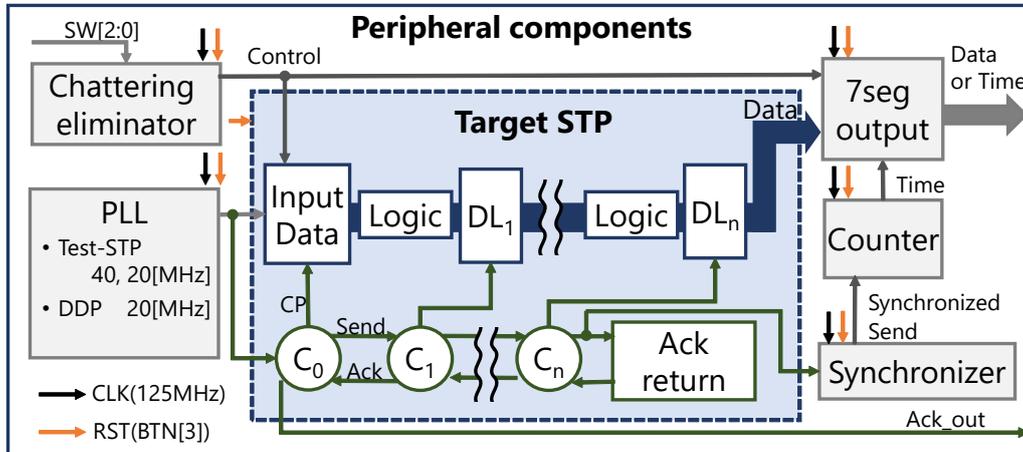


図 4.2 測定対象 STP 回路に接続する実測用周辺回路

この実測用周辺回路は 125MHz の大域クロックに合わせ動作する。また、ボードのボタン (BTN[3]) をリセット信号 RST として各モジュールに接続している。

Chattering eliminator は、ボードから入力されるスイッチ (SW[2:0]) 信号のチャタリングを除去し、control 信号として出力するモジュールである。

PLL は、Vivado の IP で提供されているクロック分周モジュールである。125MHz のクロック信号を分周して出力する。本研究の実装評価にあたっては、テスト用 STP 回路に接続する際は 40MHz または 20MHz を出力し、DDP を接続する際は 20MHz を出力している。

Synchronizer は、TargetSTP の最終段の C 素子から出力された Send 信号を 2 段カスケード接続した D-フリップフロップで同期化するモジュールである。同期化された Send 信号は Synchronized Send として Counter に送る。

Counter は、Send 信号を 100 回検出するのに要した時間を計測し、Time 信号として出力する。Send 信号の検出には、Synchronized Send を用いる。C 素子の $T_f + T_r$ の動作確認に 100 回検出した時間を用いる理由としては、C 素子の $T_f + T_r$ を直接計測しようとする、STP 回路内のすべての C 素子を監視しなくてはならないため、現実的でないからである。非同期回路であるため、“〇〇回検出した時間”はばらつくが、ばらつきの許容範囲を

4.4 ZyboZ7-20 の P_{LUT} , P_{Reg} 取得

定めておくことで、許容範囲であれば問題なく実装できていると判断でき、許容範囲を超えていれば、回路実装時に何らかの問題が発生していると判断できる。

7seg output は、出力信号を制御するモジュールで、STP 回路から出力されるデータと、Counter から出力される Time を Control で切り替えて出力する。データを出力する際、STP 回路から出力されたデータは非同期信号であるため、Synchronized Send 用いて、データも同期化する。同期化されたデータまたは、Time 信号は、7seg-LED に出力する。また、同期化されたデータは、人の肉眼で観測するには、高速で変化するため、Control 信号、つまり SW 入力で、7seg の更新と停止を切り替えられるようにし、データが正しく出力されているかを確認できるようにした。

4.4 ZyboZ7-20 の P_{LUT} , P_{Reg} 取得

本研究では TEG 回路として、32bit のデータを処理する非同期パイプライン回路を使用する (図 4.3)。

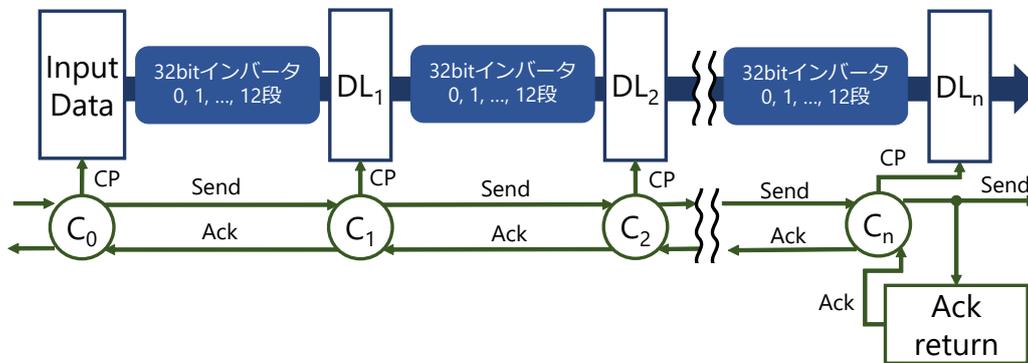


図 4.3 テスト用 STP の構成

Logic には線状に接続したインバータを使用し、DL の動作制御は C 素子が行う。回路パターンを入力レートを f として：20, 40MHz, DL 段数:50, 100 段, Logic のインバータ 0, 2, 4, 6, 8, 6, 10, 12 段, 計 28 パターンのテスト用 STP 回路の電力を実測する。

Logic のインバーター段数が LUT の数に影響し、DL 段数は Register の数と LUT の数に影響する。この回路は Logic が Not を取りつづけるため、LUT と DL の動作確率は

4.4 ZyboZ7-20 の P_{LUT} , P_{Reg} 取得

常に 1 である。回路内の InputData はパイプライン回路に bit を反転させた 32bit 信号 0xaaaa_aaaa, 32bit 信号 0x5555_5555, 2つの値を交互に送信する。

また, C 素子の $T_f + T_r$ を入力レート f と一致させるのは難しいので, 各 C 素子の遅延素子数を調節し, 各 C 素子の $T_f + T_r$ を $1/f - 5[ns]$ 以内に収めている。

28 通りのテスト用 STP 回路を FPGA に実装するにあたり, 各試行で回路パターン以外の条件を揃える必要がある。Vivado を用いた FPGA 設計工程では, 論理合成後の配置配線で, 通常は, Vivado の自動配置配線機能により配置配線が行われる。しかし, この機能はわずかに RTL を書き換えるだけで配置配線結果が大きく変化する可能性がある。テスト用 STP 回路においては, C 素子の $T_f + T_r$ が変化する可能性があり, そうなれば実測結果に影響する。

本研究では対策として, LOC・BEL 属性による回路配置の固定と PBlock 属性によるモジュールのエリア分けを行った。これらの属性は, 制約ファイルに配置配線時の制約として記述して使用する。Vivado は論理合成, 配置配線, FPGA 実装時に制約ファイルを参照し, そのファイル内の制約に従うため, 制約ファイルに記述した属性によりこれらの工程内の処理をある程度制御できる。

LOC 属性はモジュールが使用する領域を BLE Slice 単位で座標を指定するもので, BEL 属性は, Slice 内の何番目の LUT, Register を使用するかを指定する。LOC・BEL 属性は合わせて使用することで, “モジュールを座標○○にある BLE Slice の△△番目の LUT に配置する”といったように指定することができる。

LOC・BEL 属性は事実上の配置の固定であり, Vivado の自動配置配線のうち自動配置の影響をなくすことができる。ただし, 配線は変化する場合がある。図 4.4 に示すように, テスト用 STP 回路においては, 最も回路規模が大きくなる回路パターンである DL100 段, インバーター 12 段のテスト用 STP 回路先に実装し, LOC・BEL 属性による固定を行った後, 電力実測する。次に, 2 番目に回路規模が大きい回路パターンである DL100 段, インバーター 10 段のテスト用 STP 回路を実装すると, 配置が前のテストパターンと同じになる。差分であるインバータ 11 段目, 12 段目以外の回路はモジュールが存在しないので影響しない。

4.4 ZyboZ7-20 の P_{LUT} , P_{Reg} 取得

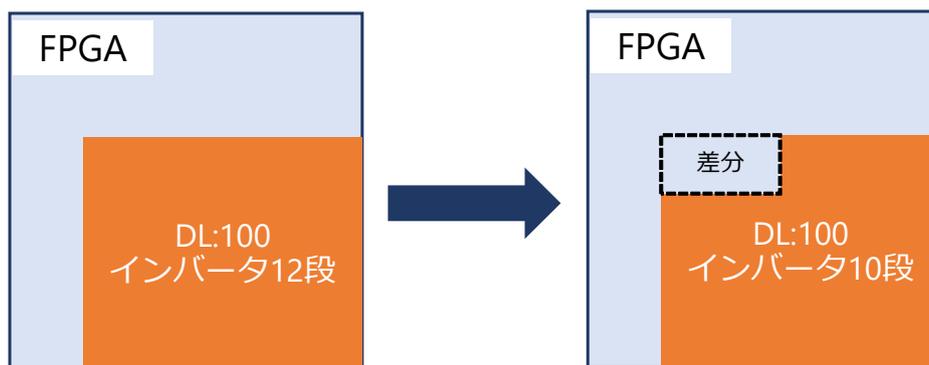


図 4.4 LOC・BEL による配置固定のイメージ

PBlock 属性は, FPGA 上に PBlock エリアを作成し, そこに論理合成後のモジュールを割り当てる. テスト用 STP 回路におけるモジュールとは, C 素子, Logic(インバータ)・DL や実装用周辺回路のことである. 割り当てられたモジュールは配置配線時に, 指定された PBlock エリアの中に配置される.

PBlock エリアのエリア分けを図 4.5 に示す.

FPGA 上を C 素子, Logic・DL, 実装用周辺回路の 3つのエリアに分割した. FPGA の中央に赤色で示しているのが C 素子の PBlock エリア, その周りを取り囲むように青色で示しているのが Logic・DL の PBlock エリアである. そして, 緑色で示すエリアに入出力のために接続している実測用周辺回路用の PBlock エリアを配置している. 図中のオレンジ色の線はデータの流を表しており, データは時計回りに FPGA を一周する.

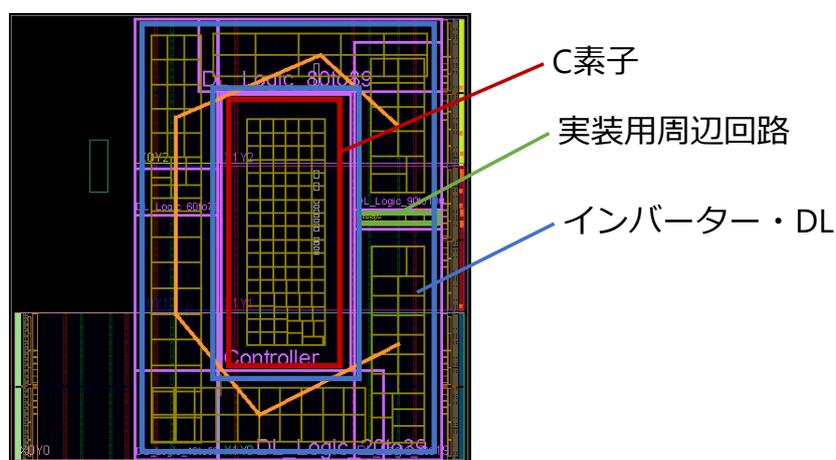


図 4.5 テスト用 STP 回路の PBlock エリア分け

4.4 ZyboZ7-20 の P_{LUT} , P_{Reg} 取得

実測用周辺回路の PBlock エリアはデータの流れの始点と終点の間に位置している。実測用周辺回路の PBlock エリアをここに置いてある理由は、FPGA とボードを接続している Input/OutputPIN が FPGA の右端に配置されており、その近くに入出力を担う実測用周辺回路を配置するためである。

C 素子と Logic・DL の PBlock エリアを分けているのは、C 素子、Logic、DL がランダムに配置されてしまうと Logic・DL の回路パターン変更に伴って C 素子の $T_f + T_r$ が変化してしまう可能性があるためである。LOC・BEL 属性を用いて配置の固定をした場合でも配線は固定できないため、C 素子と Logic・DL がランダムに配置されていると Logic・DL の回路パターンを変更に伴って C 素子間の配線も変更されてしまう。それにより、C 素子の配線遅延が変化し、結果として $T_f + T_r$ が変動する。これは、実測電力に影響が出る可能性がある。C 素子と Logic・DL 用の PBlock エリアを分けることで、Logic・DL の回路パターンを変更した際の配線変化の影響が Logic・DL 用の PBlock エリアの中に納まるようになり、C 素子にまで及ばなくなる。

これらを踏まえて、テスト用 STP 回路の実測を 1. $f=20\text{MHz}$ ・DL100 段・インバータ 0~12, 2. $f=40\text{MHz}$ ・DL100・インバータ 0~12 段, 3. $f=20\text{MHz}$ ・DL50 段・インバータ 0~12, 4. $f=40\text{MHz}$ ・DL50 段・インバータ 0~12 の順に以下に示す流れで実施した。周波数を変更するときは C 素子の配置固定を解除して、遅延量を調節している。インバータ・DL は常に固定した。

1. C 素子、Logic・DL、実測用周辺回路、それぞれの PBlock エリアを PBlock 属性を使って作成
2. 最も大きい回路パターン (20MHz・DL100 段・インバータ 12 段) を実装
3. LOC・BEL 属性を使って配置を固定
4. 電力を実測
5. 次に大きい回路パターンを実装 (例えば、20MHz・DL100 段・インバータ 12 段の次に回路規模が大きいのは、20MHz・DL100 段・インバータ 10 段)

4.4 ZyboZ7-20 の P_{LUT} , P_{Reg} 取得

6. 電力を実測

7. 5 と 6 をインバータ 0 段まで繰り返す

Zynq7020 $f=20\text{MHz}$ ・DL100 段・インバータ 12 段, $f=40\text{MHz}$ ・DL100 段・インバータ 12 段, $f=20\text{MHz}$ ・DL50 段・インバータ 12 段, $f=40\text{MHz}$ ・DL50 段・インバータ 12 段を例に, 配置配線後の Zynq7020 上の配置レイアウトを図 4.6 と図 4.7 に示す.

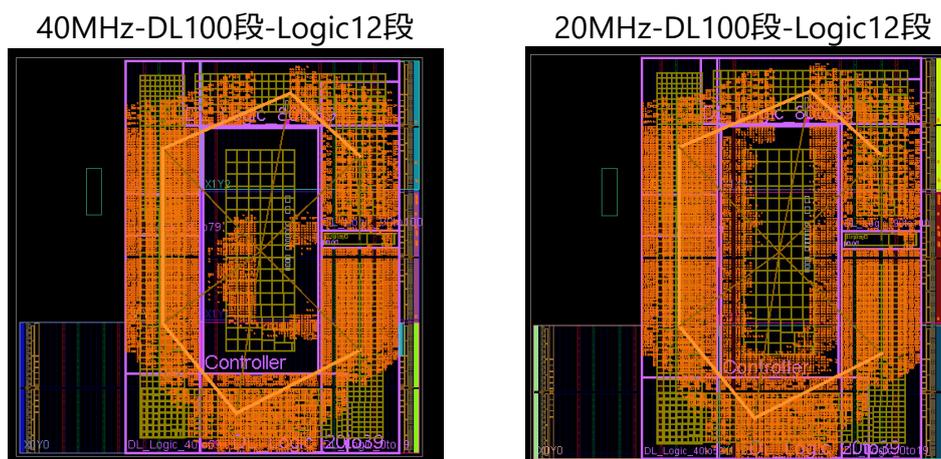


図 4.6 テスト用 STP 回路 DL100 段, インバータ 12 段の配置レイアウト

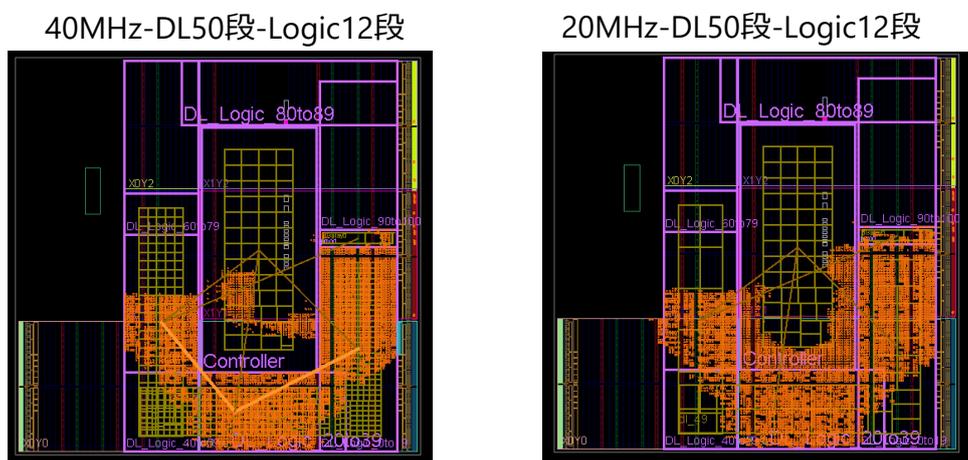


図 4.7 テスト用 STP 回路 DL50 段, インバータの配置レイアウト

テスト用 STP 回路を ZyboZ7-20 に実装後, 実測電力と設計時に決定した f , 実装時の N_{LUT} , N_{Reg} をいくつかの回路パターンのみ抜粋して表 4.1 に示す. すべてのデータは付録 A に記載する.

4.4 ZyboZ7-20 の P_{LUT} , P_{Reg} 取得

表 4.1 テスト用 STP の各回路パターンと実測電力

DataID	f [MHz]	DL 段数	インバータ段数	N_{LUT}	N_{Reg}	実測電力 [W]
1	20	100	12	31,235	3,334	1.628
2	20	100	10	28,508	3,334	1.620
8	40	100	12	27,183	3,334	1.751
9	40	100	10	25,985	3,334	1.723
15	20	50	12	15,328	1,734	1.553
16	20	50	10	14,371	1,734	1.545
22	40	50	12	13,603	1,734	1.620
23	40	50	10	13,015	1,734	1.590

テストSTP回路の実測電力, 各 f , N_{LUT} , N_{Reg} を用いてモデル A, モデル B に従って 28 個のデータの回帰分析を行った. 各回路パターンの実測電力を Y 軸, N_{LUT} を X 軸としたグラフを図 4.8 に示す. グラフ内の直線は, モデル A の回帰分析に合わせて近似した結果を示す近似直線である. グラフより実測電力は N_{LUT} に対して右肩上がりになっており, 比例の関係にあると推測できる.

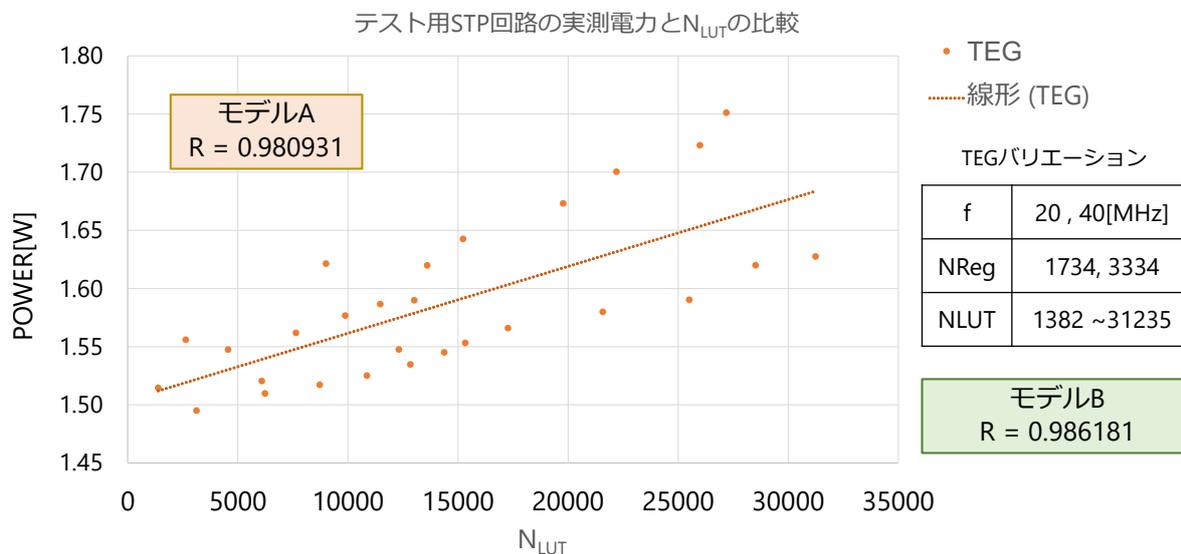


図 4.8 実測電力とモデル A の解析結果のグラフ

また, グラフ内には提案モデル A と提案モデル B の相関係数 R を記載している. 相関係

4.4 ZyboZ7-20 の P_{LUT} , P_{Reg} 取得

数 R より, TEG 回路のパラメータ f , N_{LUT} , N_{Reg} と実測電力に高い相関があることが分かる.

4.4 ZyboZ7-20 の P_{LUT} , P_{Reg} 取得

4.4.1 提案モデル A の P_{LUT} , P_{Reg} の取得

提案モデル A では、(動作時電力) = ((実測電力) - (STP 停止時電力))/ f より求めた動作時電力を目的変数とし、説明変数を $N_{LUT}:X_1$, $N_{Reg}:X_2$ として、2 変数重回帰分析を行った。STP 停止時電力は回路パターン $f=20 \cdot \text{DL50}$ 段・インバータ数 0 段の STP 停止時電力 1.490W が全回路パターンの中で最も小さかったのでこれをテスト用 STP における STP 停止時電力とした。2 変数重回帰分析から得られた回帰統計を表 4.2 に示す。決定係数 R^2 または、補正決定係数 R^2 は、目的変数を説明変数でどれだけ説明できているかを示すものであるため、TEG 回路として、テスト用 STP 回路を用いた電力の実測において、変更した回路パターン以外のノイズをほとんど抑制できていること推測できる。

表 4.2 提案モデル A に基づく 2 変数重回帰分析の回帰統計

重相関 R	重決定 R^2	補正 R^2	標準誤差	観測数
0.980	0.962	0.959	3.76×10^{-10}	28

次に、切片, $X_1(N_{LUT})$, $X_2(N_{Reg})$ の回帰係数を表 4.3 に示す。

表 4.3 提案モデル A の切片と説明変数 $X_1(N_{LUT})$, $X_2(N_{Reg})$ の回帰係数

	係数	標準誤差	t	p	上限 95%	下限 95%
切片	-7.005×10^{-10}	2.362×10^{-10}	-2.964	0.006	-1.18×10^{-9}	-2.13×10^{-10}
X_1	1.899×10^{-13}	1.059×10^{-14}	17.932	8.719×10^{-16}	1.68×10^{-13}	2.11×10^{-13}
X_2	4.632×10^{-13}	1.084×10^{-10}	4.275	0.0002	2.39×10^{-13}	6.86×10^{-13}

以上より、テスト用 STP 回路の実測とその結果の回帰分析から取得したモデル A における P_{LUT} , P_{Reg} , P_{others} を以下に示す。

$$P_{LUT} = 1.899 \times 10^{-13}, P_{Reg} = 4.632 \times 10^{-13}, P_{others} = -7.005 \times 10^{-10} + 1.490$$

4.4 ZyboZ7-20 の P_{LUT} , P_{Reg} 取得

4.4.2 提案モデル B の P_{LUT} , P_{Reg} の取得

提案モデル B では、目的変数を実測電力とし、説明変数を $f \times (N_{LUT} + N_{Reg}) : X_1$ として単変数回帰分析を行った。単回帰分析から得られた回帰統計を表 4.2 に示す。提案モデル A と同様に決定係数 R^2 または、補正決定係数 R^2 より、変更した回路パターン以外のノイズをほとんど抑制できていると推測できる。

表 4.4 提案モデル B に基づく単回帰分析の回帰統計

重相関 R	重決定 R^2	補正 R^2	標準誤差	観測数
0.986	0.972	0.971	0.011	28

次に、切片, X_1 の回帰係数を表 4.5 に示す。

表 4.5 提案モデル B の切片と説明変数 X_1 の回帰係数

	係数	標準誤差	t	p	上限 95%	下限 95%
切片	1.480	0.004	366.4	8.31×10^{-50}	1.472	1.489
X_1	2.132×10^{-7}	7.025×10^{-9}	30.3	7.87×10^{-22}	1.98×10^{-7}	2.27×10^{-7}

以上より、テスト用 STP 回路の実測とその結果の回帰分析から取得した提案モデル B における P_{LUT} , P_{Reg} , P_{others} を以下に示す。

$$P_{LUT} = P_{Reg} = 2.132 \times 10^{-7}, P_{others} = 1.480$$

4.5 DDP 搭載 ZyboZ7-20 電力推定モデルの評価

検証のために提案モデル A と提案モデル B で推定した電力と、DDP 搭載 ZyboZ7-20 の実測電力の比較評価を行った。電力推定モデルによる電力推定は、デバイスの最大電力や実用時の電力等いくつかの場合分けが考えられる。ここでは、実用時の電力推定を対象に電力推定を行い、評価する。比較評価にあたり、DDP1 コアでは電力が小さく、評価に影響する可能性を考慮して、本研究では、DDP を 4 コア搭載した ZyboZ7-20 を対象に電力の推定と実測を行って比較する。

検証用 DDP はデータ部が 16bit, 32bit の 2 通りの DDP アーキテクチャを使用して検証する。それぞれを 16bitDDP, 32bitDDP と呼称する。DDP プログラムには、実用的なプログラムと、TEG 回路のように簡単な計算を行うプログラムを使用する。

実用化された DDP 搭載 FPGA ボードの用途先としてセンサから受信したデータを FIR によってフィルタ処理することが挙げられる。そこで、これを検証用 DDP で実行する DDP プログラムとして、検証用 DDP への入力データには音声信号を使用した。

簡単な計算を繰り返すプログラムでは、待ち合わせをせず、定数メモリから定数を読み出し、入力パケットと定数メモリの出力を乗算した結果を、DDP 内を周回/パケット複製などをせずそのまま出力する。

4.5.1 FIR-DDP プログラム

FIR-DDP プログラムの詳細を示す。このプログラムは 13 タップの FIR フィルタで、固定小数点数を対象に演算を行う。作成した DDP プログラムを図 4.9 に示す。

本プログラムは、入力を受けると、入力データ 12 回繰り返しコピーして 13 個のデータとし、各データをそれぞれ b_0 から b_{12} の係数を適用する。このとき、乗算命令 mul と 1bit 左シフト命令 Lshift を使用する。そして、それらをすべての総和をとる。この時 add 命令を使用する。

DDP ではデータが待ち合わせが必要なデータが揃うと発火し、待ち合わせが成立する。

4.5 DDP 搭載 ZyboZ7-20 電力推定モデルの評価

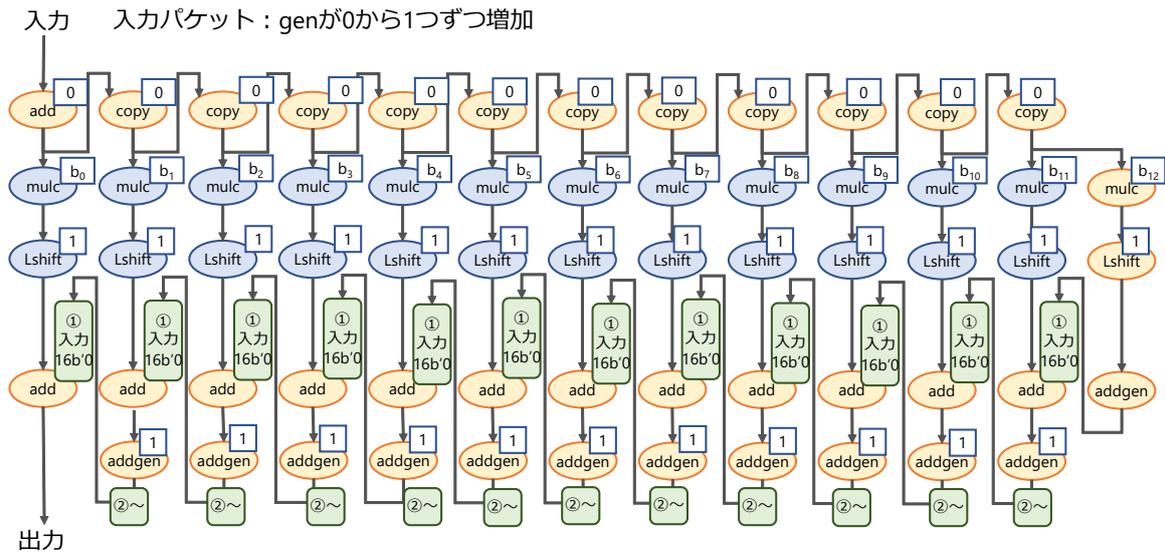


図 4.9 FIR-DDP プログラム

このとき、前述したようにヘッダ情報をもとに待ち合わせを行う。ここで、入力データ n_1 , n_2 , n_3 , n_4 のヘッダ情報が同じだと、入力データ n_1 の入力データ n_2 の待ち合わせが成立し、入力データ n_3 と入力データ n_4 の待ち合わせが成立してほしいところを、入力データ n_1 のパケットと入力データ n_3 の待ち合わせが成立し、入力データ n_2 のパケットと入力データ n_4 の待ち合わせが成立するといった事態が起こりうる。これでは正しい演算結果を得られないので、対策が必要である。

本プログラムでは、パケットのヘッダ部の generation フィールドの値を 1 ずつずらし、同じ generation 同士で待ち合わせするようにすることでこの問題をパスしている。そのために、入力パケットのヘッダ情報内の generation フィールドを 1 ずつ加算して入力し、DDP プログラム中では、generation フィールドの値に 1 を加算している。検証用 DDP のアーキテクチャでは、generation フィールドは -128~127 の値をとるように設計していたので、入力パケットと DDP プログラム内共に 0~127 の間をループさせている。

図中の緑色枠内の①入力 16b'0 とあるように、本 DDP プログラムをアプリケーションとして使用する際、係数適用後の総和をとる工程で、初期値となる値が 0 のデータを待ち合わせデータとしてあらかじめ DDP の MM に待機させておく必要がある。そのため、DDP をリセット後、まず 12 個の初期値を入力パケットとして、DDP に入力し、その後、音声デー

4.5 DDP 搭載 ZyboZ7-20 電力推定モデルの評価

タを含むパケットを入力した。この初期値が必要なのは、DDP リセット後、最初の入力パケットのみで、その後は、係数適用後のデータが待ち合わせ対象として待機するので、12 個の初期値の入力は DDP リセット直後の 1 度のみでよい。図に示している DDP プログラムはデータ部が 16bit の DDP を対象にしているので、データ部が 32bit の DDP に用いる場合は初期値を 32b'0 にする必要がある。

本プログラムでは、パケットが入力されると 62 個のノードの演算が実行される。パケットの入力周期は、この演算処理が終わる時間以上にする必要がある。検証用 DDP の $T_f + T_r$ を 50ns と見積もると、ノードの演算処理にかかる時間は 3000ns 強である。本研究では、パケットの入力周期を 5050ns に設定した。

4.5.2 検証用 DDP 実装・電力実測

提案モデルで推定した電力と比較するために検証用 DDP の電力実測は、テスト用 STP 回路の実測時と同様の手順で行った。PBlock エリアもテスト用 STP 回路と同様に、C 素子を中心に周囲を各ステージが取り囲むようなレイアウトに設定した。16bitDDP と 32bitDDP で回路規模に大きな差がなかったため、両者で同じ PBlock エリアを設定し、使用した。16bitDDP、32bitDDP を 4 コア実装した際の FPGA 上のレイアウトを図 4.10 に示す。

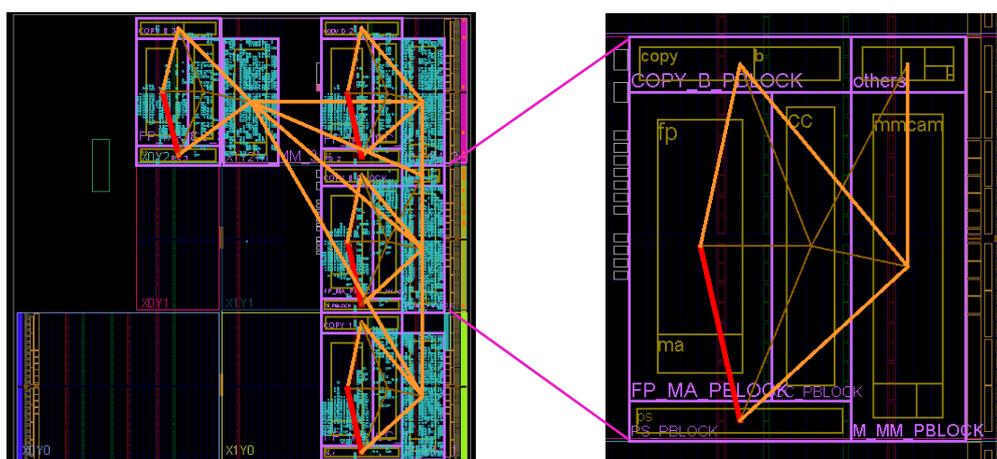


図 4.10 16bitDDP の PBlock レイアウト

4.5 DDP 搭載 ZyboZ7-20 電力推定モデルの評価

図中右側は、DDP1 コアの PBlock エリア分けである。DDP の中心に C 素子の PBlock エリアを設置し、それを取り囲むように M・MMCAM・MMRAM ステージの PBlock エリア、PS ステージの PBlock エリア、FP・MA ステージの PBlock エリア、COPY・B ステージの PBlock エリア、実測用周辺回路の PBlock エリアを用意した。

図中左側は、Zynq7020 上の DDP4 コアの PBlock エリアと実際の配置配線結果を示している。4つの DDP コアは図中のように設定した。青色で示されているのが配置配線結果の配置レイアウトである。オレンジ色・赤色の線は、DDP 内のデータの接続関係を示していると考えられる。

次に、実測した各 DDP の電力を表 4.6 に示す。結果より、16bitDDP4 コアと 32bitDDP4 コアの実測電力に大きな差はないことが分かる。

表 4.6 16bitDDP と 32bitDDP の実測電力

単位 [W]	DDP Data:16bit	DDP Data:32bit
FIR-Program[W]	1.50579	1.51876
Simple-Program[W]	1.59005	1.61444

4.5.3 提案提案モデル A と提案電力推モデル B を用いて推定電力を導出

各 DDP の N_{LUT} , N_{Reg} は、実測時に Vivado で調べたものを使用する。

各ステージの f は、入力レートあたりの動作回数をプログラムから算出した。FIR プログラム動作時のデータの入力レートは 5,050ns とした。まず、M, MMCAM, MMRAM, ステージは、入力毎に 74 パケットが通過するため動作周期は $5050[ns]/74[パケット] \doteq 68.243[ns]$ として、動作周波数は約 14.653MHz とした。

次に、PS, FP, MA ステージは、MMRAM ステージでパケット 12 個のパケットがマッチングするため、PS ステージを通過するパケットが 12 パケット減り、62 パケットとなるため、動作周期は $5050[ns]/(74 - 12) \text{パケット} \doteq 81.243[ns]$ として、動作周波数は約 12.277MHz とした。

4.5 DDP 搭載 ZyboZ7-20 電力推定モデルの評価

そして、COPY, B ステージは、FP ステージから COPY ステージに行くパケットのうち 12 パケットがコピーを行うため、COPY ステージと B ステージは FP ステージに比べ通過パケットが 12 個多くなるため、その動作周期は、 $5050[ns]/(62 + 12)[\text{パケット}] \doteq 68.243[ns]$ 動作周波数は約 14.653MHz とした。

加えて、このプログラムはデータメモリへのアクセスを行わないため、MA ステージの DL は動作するが、Logic は動作しない。よって、MA ステージの α_{LUT} を 0 とした。ほかのステージの α_{LUT} 及び α_{Reg} については、入力されたパケットや、FIR の係数により変動しうるが、詳細に求めることは現実的でないため、ここでは、 α_{LUT} , α_{Reg} を 1 とし、最も消費電力が大きくなる場合の電力を推定した。

また、簡単なプログラムでは、DDP プログラムの構造や入力周期で変化しうる f , α_{LUT} , α_{Reg} について検討した。簡単なプログラムでは、パケットの入力周期を 50ns とした。パケットの待ち合わせ、複製、削除を行わないので、各ステージの動作周期も 50ns である。よって、 f を 20MHz とした。 α_{LUT} , α_{Reg} は FIR プログラムの時と同様に 1 とした。

最後に、FIR-DDP プログラムと簡単なプログラムを実行した際の推定電力を各 DDP の N_{LUT} , N_{Reg} , f , α_{LUT} , α_{Reg} と、回帰分析より得た各モデルの P_{LUT} , P_{Reg} を用いて、提案モデル A と提案モデル B の推定電力を算出した。

DDP プログラムが異なる場合でも、 f , α_{LUT} , α_{Reg} が異なるだけで、導出過程は同じであるため、ここでは、FIR-DDP プログラムを実行時の 16bitDDP の各パラメータと各ステージの推定電力、DDP 全体の電力を表 4.7 に示し、FIR-DDP プログラムを実行時の 32bitDDP の各パラメータと各ステージの推定電力、DDP 全体の電力を表 4.8 に示す。簡単なプログラムの推定電力は結果のみを後述する。

表には各ステージの f , α_{LUT} , α_{Reg} , N_{LUT} , N_{Reg} をロジック・DL と C 素子で分けて記載している。そして、両モデルで各ステージのロジック・DL・C 素子の電力をまとめて推定し、その推定値を記載している。そして、算出した DDP 単体の推定電力、DDP4 コア分の電力、DDP4 コアと P_{Others} を合計した DDP 搭載 ZyboZ7-20 の推定電力を記載している。

4.5 DDP 搭載 ZyboZ7-20 電力推定モデルの評価

表 4.7 16bitDDP の各パラメータと各モデルの推定電力

	f [MHz]	α_{LUT}	N_{LUT}	α_{Reg}	N_{Reg}	Model-A[W]	Model-B[W]
M	14.653	1	28	1	76	0.00085	0.00061
(CM)	14.653	1	92	1	0	—	—
MMCAM	14.653	1	745	1	1,326	0.01117	0.00658
(C)	14.653	1	35	1	0	—	—
MMRAM	14.653	1	95	1	38	0.00063	0.00053
(CE)	14.653	1	35	1	1	—	—
PS	12.277	1	1	1	52	0.00038	0.00023
(C)	12.277	1	35	1	0	—	—
FP	12.277	1	658	1	62	0.00196	0.00196
(CE)	12.277	1	28	1	1	—	—
MA	12.277	0	61	1	41	0.00048	0.00011
(C)	12.277	1	44	1	0	—	—
COPY	14.653	1	15	1	40	0.00053	0.00038
(CX2)	14.653	1	60	1	8	—	—
B	14.653	1	0	1	38	0.00036	0.00023
(CB)	14.653	1	34	1	1	—	—
DDP	—	—	1,987	—	1,722	0.01636	0.01063
DDP4Core	—	—	—	—	—	0.06544	0.04253
DDP+ P_{others}	—	—	—	—	—	1.55544	1.52347

4.5 DDP 搭載 ZyboZ7-20 電力推定モデルの評価

表 4.8 32bitDDP の各パラメータ

	f [MHz]	α_{LUT}	LUT	α_{Reg}	Register	Model-A[W]	Model-B[W]
M	14.653	1	44	1	108	0.00110	0.00075
(CM)	14.653	1	89	1	0	—	—
MMCAM	14.653	1	745	1	1,342	0.01127	0.00662
(C)	14.653	1	32	1	0	—	—
MMRAM	14.653	1	162	1	54	0.00091	0.00077
(CE)	14.653	1	29	1	1	—	—
PS	12.277	1	1	1	84	0.00056	0.00031
(C)	12.277	1	35	1	0	—	—
FP	12.277	1	1,344	1	94	0.00375	0.00385
(CE)	12.277	1	31	1	1	—	—
MA	12.277	0	83	1	57	0.00062	0.00015
(C)	12.277	1	42	1	0	—	—
COPY	14.653	1	15	1	56	0.00063	0.00042
(CX2)	14.653	1	54	1	8	—	—
B	14.653	1	0	1	54	0.00047	0.00028
(CB)	14.653	1	35	1	1	—	—
DDP	—	—	2,757	—	1,914	0.01930	0.01315
DDP4Core	—	—	—	—	—	0.07721	0.05260
DDP+ P_{others}	—	—	—	—	—	1.56721	1.53354

4.6 考察

FIR-DDP プログラムと、簡単な DDP プログラムを実行時の 16bitDDP と 32bitDDP の電力について、提案モデル A の推定値とモデル B の推定値と実測値を比較した表を 4.9 に示す。誤差は平均絶対百分率誤差より求めた。提案モデル A は平均絶対百分率誤差が 0.488%~3.297%，提案モデル B は平均絶対百分率誤差が 0.9734%~3.407%であることを確認した。両モデル共に、実行する DDP プログラムによって推定精度が変化することが分かる。DDP プログラムで異なっていたのは f であり、 f がモデルの推定精度に大きく影響される可能性が示唆された。

表 4.9 2つのモデルによる推定値と実測値の比較

単位 [W]	Model-A	Model-B	実測値	モデル A 誤差	モデル B 誤差
FIR Data:16bit	1.55544	1.52347	1.50579	3.297%	1.174%
FIR Data:32bit	1.56721	1.53354	1.51876	3.190%	0.973%
Simple Data:16bit	1.58229	1.54320	1.59005	0.488%	2.946%
Simple Data:32bit	1.60059	1.559425	1.61444	0.857%	3.407%

4.6 考察

本研究では、実証検証に、DDP プログラムでは、FIR プログラムと乗算を行う簡単なプログラムを用い、DDP アーキテクチャには 16bitDDP と、32bitDDP を用いたが、より詳細な評価のためには今後、異なる構造を持つ DDP プログラムや DDP アーキテクチャ等、異なる条件での推定精度の検証が必要である。より具体的な検証のために、応用分野に特化した DDP アーキテクチャを検証に用いることも重要である。加えて、DDP 搭載 FPGA ボードを IoT デバイスとして使用するにあたり、FIR フィルタで処理するデータが音声である可能性は低い、そのため、音声データ以外の異なるデータを入力データとしたときの電力を精度評価に用いるのがより望ましい。IoT エッジデバイスとしてどのようなデータが入

4.7 結言

力されるのかという検討を含めて、これについては今後の課題とする。

Zybo Z7-20 では、DDP を実装した FPGA の電力がボードの電力 P_{others} に比べ小さく、精度が高くなったが、異なる FPGA ボードで同様の水準の精度が得られるか検証が必要である。

DDP 各ステージの動作周波数 f や、動作確率 α_{LUT} , α_{Reg} については、DDP アーキテクチャ、DDP プログラム構造で変化するため、DDP の扱いに慣れた回路設計者でないと動作確率の導出は難しい。そのため、今後、DDP アーキテクチャ、DDP プログラム構造を基に自動的、もしくは半自動的に動作確率を導出する手法の検討が必要である。

電力モデルに基づく電力推定方法は、デバイス固有の電力特性を使用するため、事前のデバイスの電力特性調査をデバイス毎に行う必要がある。電力モデルに基づく電力推定法を活用するにあたり、TEG 回路の妥当性を検討する必要がある。さらに、デバイスの電力特性調査にかかる工程・手間・時間がかからないことが本提案手法をより便利にするためには重要である。本研究では、TEG 回路の回路パターンを 28 通り用意して電力を実測したが、回路パターン数により、推定精度は変動すると考えられる。デバイスの電力特性調査にかかる工程や手間、時間を短縮をできないか改良する必要があり、回路パターンを減らした際の精度への影響の検証が必要である。

4.7 結言

本章では、前章で提案した FPGA 電力推定モデルに基づく電力推定法の実装評価を実施した。実証評価にあたって、まず、実装用周辺回路の説明を行った。そして、ZyboZ7-20 の P_{LUT} , P_{Reg} を取得するために用いる TEG 回路であるテスト用 STP 回路を説明し、その実装手順を説明した。実装後、実測した電力は提案モデル A とモデル B に従って回帰分析を実施した。最後に、検証用 DDP をテスト用 STP 回路と同様の手順で実装し、検証用 DDP の N_{LUT} , N_{Reg} と導出した f , α_{LUT} , α_{Reg} と、事前に取得した P_{LUT} , P_{Reg} をもとに電力を推定し、実測した電力と比較した結果、提案モデル A は平均絶対百分率誤差が

4.7 結言

0.488%~3.297%, 提案モデル B は平均絶対百分率誤差が 0.973%~3.407%であることを確認した.

第 5 章

結論

本研究では、IoT デバイスとして有望な DDP 搭載 FPGA ボードの電力推定法として、DDP 搭載 FPGA ボードの電力推定モデルを仮定し、モデルに基づいて電力推定を行う手法の提案を行った。

はじめに、電力推定モデルの仮定に際しては、DDP は非同期回路であるため、各ステージで回路構成、動作周波数、動作確率が異なることを説明した。また、各ステージの電力を積算して DDP 全体の電力を求める必要があることを説明し、DDP を FPGA 実装した際に電力に影響する要素として、回路構成、動作周波数、動作確率、FPGA デバイスの電力特性が各ステージの電力にどのように影響するかを説明した。各ステージの回路構成としては、LUT と Resister が支配的であり、主要な電力推定パラメータであるため、それぞれの数を N_{LUT} , N_{Reg} と定義した。動作周波数 f は、C 素子の $T_f + T_r$ とアプリケーションの動作周期、DDP プログラム構造により決定し、LUT と Register 動作確率 $\alpha_{LUT}, \alpha_{Reg}$ は DDP プログラム構造と DDP プログラムにより決定することを説明した。また、FPGA デバイスの電力特性は非公開データであるため、これを取得するために TEG 回路を用いて事前に電力を実測し、TEG 回路の各パラメータ N_{LUT} , N_{Reg} , f , $\alpha_{LUT} = \alpha_{Reg} = 1$ と実測電力の関係を回帰分析により解析した結果を FPGA デバイスの電力特性を示す不変的なパラメータ P_{LUT} , P_{Reg} とし、電力推定モデルに用いることを説明した。

そして、これらのパラメータを用いて基礎となる電力推定モデルを定式化し、モデルに基づいて各ステージの電力を算出し、すべてのステージの電力の総和とその他の電力から DDP 搭載 FPGA ボードの電力を算出することを説明した。

加えて、基礎的なモデルはいくつかの解釈が可能で、本研究では 2 つの解釈として、基礎

的なモデルに忠実な電力推定モデルである提案モデル A と、LUT と Reg の電力を同じと見なすことで、推定電力の導出過程を簡潔にした電力推定モデルである提案モデル B を提案した。

最後に、提案モデル A と提案モデル B の推定精度を評価するため、AMD 社の FPGA チップである Zynq7020 を搭載した Digilent 社の FPGA ボード Zybo Z7-20 を対象に実装評価を実施した結果、提案モデル A は実測値に対して提案モデル A は平均絶対百分率誤差が 0.488%~3.297%、提案モデル B は平均絶対百分率誤差が 0.973%~3.407%であることを確認し、提案した電力推定モデルに基づく電力推定法により、ある程度の水準の精度で電力を推定できることを確認した。

今後の課題として、①さらに詳細な評価のために、異なる構造を持つ DDP プログラムや DDP アーキテクチャ等、異なる条件での推定精度の検証。② Zybo Z7-20 では、DDP を実装した FPGA の電力がボードの電力 P_{others} に比べ小さく、精度が高くなったが、異なる FPGA ボードで同様の水準の精度が得られるか検証。③ DDP アーキテクチャ、DDP プログラム構造を基に自動的、もしくは半自動的に動作確率を導出する手法の検討。④デバイスの電力特性調査にかかる工程や手間、時間を短縮をできないかの検討、回路パターンを減らした際の精度への影響の検証。これらの追加の議論や検討が残されている。

①②の解決により、本提案モデルのより詳細な精度評価が明らかになると考えられ、③④により本モデルの活用がより簡単になることが期待される。これらの課題解決により、IoT エッジデバイスとして DDP 搭載 FPGA ボードの早期の電力評価が可能になることに加えて、設計工程早期の電力評価は本提案手法を使用し、最終的なデバイスの評価には実測値を用いるなど、段階に応じた電力評価方法の使い分けが可能になる。また、これまで行われてこなかった電力評価結果を踏まえた上で設計した回路を見直す作業なども可能になる。また、電力モデルによる電力推定法は、STP-DDP に限らず、様々な非同期回路アーキテクチャへの適用も可能であると考えられるため、非同期回路搭載 FPGA ボードの電力推定法を確立できる可能性もある。

謝辞

本研究を行うにあたって、ご指導をいただいた岩田 誠 教授には深く深く感謝申し上げます。ご多忙にもかかわらず、ミーティングの時間を確保してくださり、様々な助言や指導をいただいたおかげで、研究をここまで進めることができました。加えて、研究を進めるにあたっての心構えを指導していただいたり、研究以外にも就職活動などで助言をいただいたことが非常に心強かったです。改めて、感謝いたします。

本研究の副査を引き受けてくださった横山 和俊 教授、栗原 徹 教授にも深く感謝申し上げます。お二方から貴重なご意見や疑問点を指摘していただいたおかげで、本研究を進めることができました。セミナーの時にいただいたご意見は持ち帰ったうえで研究の進め方や、次回のセミナー発表時の説明の仕方の工夫に生かさせていただきました。改めて、感謝いたします。

研究室の先輩である張 震 氏、Tamnuwat Valeeprakhon 氏には研究活動を進めるにあたってのアドバイスをいただきました。深く感謝いたします。研究室の同期である植元 陸 氏、坂口 白磨氏には心から感謝いたします。お互いの研究への意見交換は非常に勉強になりました。また、研究室運営の立場になり、二人の存在が心強かったです。

研究室の後輩である、修士1年の市ノ木 一希 氏、伊藤 雅俊 氏、岡村 健勝 氏、山下 拓 巳 氏、学士4年の大崎 綾斗 氏、奥平 舜理 氏、片岡 拓心 氏、門屋 陽丈 氏、山崎 浩正 氏、学士3年の荻田 勇人 氏には研究のご支援をいただきました。深く感謝いたします。

最後になりますが、岩田 誠 教授、横山 和俊 教授、栗原 徹 教授、研究室の先輩、同期、後輩の皆さん改めて本当にありがとうございました。そして、家族、友人にも心から感謝いたします。皆様に支えていただいたおかげで、本研究を進めることができ、本論文を執筆することができました。

参考文献

- [1] 総務省, “情報通信白書 (令和 3 年版)IoT デバイスの急速な普及,” <https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r03/html/nd105220.html>, 2025/01/14 参照.
- [2] 総務省, “情報通信白書 (令和 6 年版) エッジコンピューティング,”<https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r06/html/nd218300.html>, 2025/01/14 参照.
- [3] 高橋 龍一, “データ駆動型プロセッサの環状パイプライン構成の比較検,” 高知工科大学学士学位論文, 2022.
- [4] AMD, “Zynq-7000 SoC テクニカル リファレンス マニュアル (UG585), ”<https://docs.amd.com/v/u/ja-JP/ug585-Zynq-7000-TRM>, 2025,02/22 参照.
- [5] 松坂 拓海, “データ駆動型プロセッサの FPGA 向け電力効率向上手法の検討,” 高知工科大学学士学位論文, 2023.
- [6] Yehya Nasser, Jordane Lorandel, Jean-Christophe Prévotet, and Maryline Hélar, “RTL to Transistor Level Power Modeling and Estimation Techniques for FPGA and ASIC: A Survey,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol.40, no.3, pp. 479–493, Mar. 2021.
- [7] AMD, “Vivado Design Suite ユーザーガイド: 消費電力解析 および最適化 (UG907), ”<https://docs.amd.com/r/ja-JP/ug907-vivado-power-analysis-optimization>, 2025/01/27 参照.
- [8] AMD, “Xilinx Power Estimator ユーザーガイド (UG440), ” ”<https://docs.amd.com/r/ja-JP/ug440-xilinx-power-estimator>, 2025/01/27 参照.
- [9] Xifan Tang, Edouard Giacomini, Giovanni De Micheli, Pierre-Emmanuel, “FPGA-SPICE: A Simulation-Based Architecture Evaluation Framework for FP-

参考文献

- GAs, "IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Volume.27, no.3, pp. 637-650, Mar. 2019.
- [10] Gaurav Verma, Vijay Khare , Manish Kumar, "More Precise FPGA Power Estimation and Validation Tool(FPEV_Tool) for Low Power Applications," Wireless Personal Communications 106, pp. 2237–2246, (2019).
- [11] 岡村 健勝, "FPGA 実装した STP 回路の簡易消費電力見積り法の検討," 高知工科大学学士学位論文, 2024.

付録 A

表 A.1 テスト用 STP の各回路パターンの実測電力 (1/2)

DataID	f [MHz]	DL 段数	インバータ段数	N_{LUT}	N_{Reg}	実測電力 [W]
1	20	100	12	31,235	3,334	1.628
2	20	100	10	28,508	3,334	1.620
3	20	100	8	25,505	3,334	1.590
4	20	100	6	21,573	3,334	1.580
5	20	100	4	17,266	3,334	1.566
6	20	100	2	12,311	3,334	1.548
7	20	100	0	6,092	3,334	1.520
8	40	100	12	27,183	3,334	1.751
9	40	100	10	25,985	3,334	1.723
10	40	100	8	22,190	3,334	1.700
11	40	100	6	19,771	3,334	1.673
12	40	100	4	15,224	3,334	1.643
13	40	100	2	9,003	3,334	1.621
14	40	100	0	2,632	3,334	1.556

表 A.2 テスト用 STP の各回路パターンの実測電力 (2/2)

DataID	f [MHz]	DL 段数	インバータ段数	N_{LUT}	N_{Reg}	実測電力 [W]
15	20	50	12	15,328	1,734	1.553
16	20	50	10	14,371	1,734	1.545
17	20	50	8	12,839	1,734	1.535
18	20	50	6	10,859	1,734	1.525
19	20	50	4	8,722	1,734	1.517
20	20	50	2	6,236	1,734	1.510
21	20	50	0	3,120	1,734	1.495
22	40	50	12	13,603	1,734	1.620
23	40	50	10	13,015	1,734	1.590
24	40	50	8	11,469	1,734	1.587
25	40	50	6	9,879	1,734	1.577
26	40	50	4	7,644	1,734	1.562
27	40	50	2	4,558	1,734	1.548
28	40	50	0	1,382	1,734	1.515