

東京電機大学

博士論文

FPGA 搭載プロセッサの
高温試験に関する研究

A Study on
High Temperature Test
for Processors on FPGAs

2021 年 3 月

金子 博昭

A Study on
High Temperature Test
for Processors on FPGAs

DISSERTATION

Hiroaki Kaneko

Tokyo Denki University

March, 2021

要旨

近年、集積回路（IC）の新技术による進化は止まることを知らない。2022 年下半期には 3nm プロセスのチップ量産が始まると伝えられている。最新の IC は情報通信技術の原動力となり、情報家電や車載機器、スマートフォンなど、身近で恩恵を受ける製品への応用がますます進んでいる。これらの製品ばかりでなく、広く製品開発の短 TAT 化と、製品製造の変品種変量形態が要求されている。このような背景から、その場でプログラム可能なゲートアレイ（Field Programmable Gate Array : FPGA）のエンドユーザ向け製品への採用が拡大しつつある。一方最近では、厳しい動作環境における高信頼性が、FPGA にも要求されてきている。自動運転対応車載制御、5G 対応通信インフラ、インダストリー4.0 対応製造ライン、及び探査衛星などに向けた高信頼性製品への FPGA 採用増が要因の一つであろう。

このような状況にあって、メーカーは高温における高信頼性を保証するために、出荷に際して恒温槽によるスクリーニングを行う。また、FPGA メーカーが提供するユーザ向け開発環境には、設計保証に備えた動作速度や、消費電力、チップ温度を見積る機能の組み込みを進めている。しかしながら、実装する回路の種類・性質により、見積り精度のばらつき幅が数百倍に及ぶ場合がある。

本論文は、FPGA 加熱用プロセッサの開発、FPGA 搭載プロセッサを加熱する内部加熱器の開発と、これらを応用した研究についてまとめている。恒温槽を使用せず、加えて見積り結果を過度に依存することなく、ユーザに高温試験の環境を提供する。ユーザによる高温試験を対象とする研究は非常に少なく、低コスト FPGA 加熱方式の確立を目的として、リングオシレータの高速・高頻度動作による内部加熱能力に着目した。提案する内部加熱器は、高温試験に必要な温度が変更されても、FPGA 回路再構成で柔軟に対応できる。FPGA 構成要素として最も一般的な基本論理ブロック（CLB）と配線チャンネルを使い、単純なリングオシレータの組合せで構成するからである。

本論文は全 6 章で構成されている。第 1 章では研究の背景と目的を述べている。第 2 章では FPGA にとっての熱問題と熱利用の両面について、その背景となる IC 全般の熱問題について述べている。第 3 章では FPGA 搭載プロセッサの加熱能力について、プログラムとデータの影響を検討している。第 4 章ではリングオシレータを利用し、FPGA デバイス自体が柔軟に発熱する内部加熱器について述べている。第 5 章では高温環境下にあっても、本来のモニタ機能を維持する実装手法について述べている。第 6 章は全体

のまとめとなっている。なお、既往の研究については、各章の前半部分にまとめて記述している。以下、各章の要旨について示す。

第 1 章では、本研究の背景として FPGA に関する信頼性保証の特徴と解決すべき課題を示し、本研究の目的について述べている。加えて、最近の FPGA の特徴を確認し、背景と目的を補足している。さらに、本論文の構成と各章の概要についてまとめている。

第 2 章では、熱が FPGA にもたらす不利益な問題と、対極にある有益な利用について述べている。問題については、まず IC 全般にわたる共通点から明らかにした。利用については、FPGA 高温試験向けの温度環境を供する内部加熱器の有用性に期待している。内部加熱器は、熱により上昇するチップ温度を要因として動作速度など低下する IC 性能に着目し、FPGA に搭載するプロセッサを試験対象回路としている。

第 3 章では、内部加熱器を実装する FPGA の規模的な制約を背景として、搭載プロセッサの加熱能力について述べている。プログラムとデータの性質がおよぼす消費電力への影響、及び加熱への影響からチップ温度への関係を把握し、この関係のモデルを推定した。また、レジスタ転送レベルから新たに定義する 32bit プロセッサを FPGA に実装し、効率的にチップを加熱する発熱プログラムについてチップ温度データを収集した。さらに、4 種のプロセッサ動作速度と 5 種の発熱プログラムを組み合わせることにより、プロセッサの同期クロック信号に 1 次線形とみなせる FPGA の加熱特性を明らかにした。

第 4 章では、FPGA の高温試験を必要とする状況を背景として、リングオシレータを利用し、FPGA デバイス自体が柔軟に発熱する内部加熱器について述べている。内部加熱器は最大定格が定める温度である 125°C までのチップ加熱を目標とした。内部加熱器をプロセッサと FPGA に統合する実験環境を開発した。内部加熱器を構成する遅延素子の遅延段数 1~63 と駆動段数 63~1、活性化する加熱モジュール数最大 1,000 組とする制御パラメータを組み合わせ、チップ温度へ与える影響を解析した。内部加熱器には、これまでの研究では見られない、ゲーテッド遅延素子を多段に直列接続する拡張駆動構造とした。拡張駆動構造は、直列接続の先頭となる拡張駆動段出力の発振周波数とともに、配線路長を制御する。配線路には配線チャンネルを割り当て、金属配線とドライバを発熱に作用させる。実装には、発熱しにくい低消費電力 FPGA の中から中規模デバイスを選択した。内部加熱器を 3 種の制御パラメータを設定できるように実装し、設定の組合せがチップ温度へ与える影響を解析した。チップ温度を推奨動作温度 85°C 以上に加熱するには LUT 占有率で 29%、及び 125°C まで加熱するには LUT 占有率 54%

の FPGA リソースを使用する実装結果となった。この結果から、提案内部加熱器は高速・高性能 FPGA に実装する従来のリングオシレータと比較して、最大チップ温度、及び温度変化に対する LUT 使用数比は同一オーダーでありながら、同程度の消費電力による加熱能力を示している。

第 5 章では、高温環境下で回路が速度マージン不足となる状態を背景として、初期段階のプロセッサ開発向け機械語モニタの実装手法について述べている。提案実装手法は、プロセッサの命令セットアーキテクチャに依存しないハードウェア記述言語 (HDL) によりモニタ機能を記述し、プロセッサの HDL 記述と合流させた。FPGA に実装する機械語モニタの機能は、プロセッサ設計経験者 6 名へのアンケート回答結果によって選択した。プロセッサと統合したモニタ機能を持つ FPGA を開発し、それぞれ 100 MHz と 50 MHz の同期クロック信号を用いて動作させ、モニタ機能を正しく提供することを確認した。

最後に第 6 章では、本研究を総括するとともに、今後の展開と課題を述べている。

以上、本論文は FPGA に搭載するプロセッサの高温試験のために研究・開発した内部加熱器の構成、動作、及び作用に関して、関連研究と応用研究をまとめたものである。これらの成果は内部加熱器の応用、更にそれらを活用する製品やシステム開発への貢献が期待できる。

目次

要旨	i
目次	iv
図目次.....	vii
表目次.....	ix
第1章 序論.....	1
1.1 本研究の背景	1
1.2 本研究の目的	3
1.3 最新 FPGA の特徴	4
1.4 本論文の構成	7
第2章 FPGA における熱問題と熱利用	10
2.1 はじめに	10
2.2 IC の熱問題	11
2.3 IC の熱モデル.....	14
2.4 FPGA の熱問題	16
2.5 FPGA の熱利用	20
2.6 おわりに	22
第3章 FPGA 搭載プロセッサのプログラム実行によるチップ加熱の検討	23
3.1 はじめに	23
3.2 サブワット級 IC の熱問題.....	25
3.3 チップ加熱実験の概要	26
3.3.1 プラットホームの選択.....	26
3.3.2 温度センサの選択.....	27
3.4 実験装置の概要とターゲットマイクロプロセッサの実装	29
3.4.1 ターゲットプロセッサの命令セットアーキテクチャ	30
3.4.2 ターゲットプロセッサのマイクロアーキテクチャ.....	31
3.4.3 FPGA への実装.....	32
3.5 発熱プログラムを用いた評価.....	34
3.5.1 予備実験の結果とフィードバック	34
3.5.2 プログラム実行による発熱の基本的な挙動	36
3.5.3 発熱プログラムの概要.....	38

3.6 発熱と消費電力との関係モデル	43
3.6.1 FPGA デバイスの消費電力算出方式	43
3.6.2 消費電力と関連温度検討の課題	43
3.7 おわりに	46
第4章 FPGA チップ加熱用リングオシレータの実装と評価	47
4.1 はじめに	47
4.2 リングオシレータと関連研究	48
4.3 拡張駆動リングオシレータ加熱器の提案	50
4.3.1 FPGA の進化と懸案事項	50
4.3.2 IC の熱発生	51
4.3.3 リングオシレータ加熱器	51
4.4 FPGA への実装	52
4.4.1 ゲーテッド遅延素子	52
4.4.2 ROH モジュール	54
4.4.3 加熱評価システム	55
4.4.4 温度センサ	58
4.4.5 FPGA の選定	59
4.4.6 実装結果	62
4.5 チップ加熱実験	64
4.5.1 予備実験の結果	64
4.5.2 参照 ROH モジュール	66
4.5.3 発振周波数の制御	67
4.5.4 加熱の制御	68
4.5.5 加熱の均一性	73
4.5.6 実験結果と考察	76
4.6 チップ温度制御への応用	78
4.7 おわりに	81
第5章 FPGA 搭載プロセッサ向け機械語モニタのハードウェア実装	82
5.1 はじめに	82
5.2 プロセッサの開発環境	84
5.3 提案する機械語モニタ	87
5.3.1 構成概要	87
5.3.2 動作概要	88
5.3.3 設計手法の概要	89
5.4 単体実装実験と評価	92
5.4.1 実験プラットフォーム	92

5.4.2 モニタ機能の動作確認.....	94
5.4.3 実装結果.....	96
5.5 プロセッサ統合への検討.....	99
5.5.1 統合するプロセッサの要件.....	99
5.5.2 統合の貢献対象と貢献時期.....	99
5.5.3 プロセッサ設計者へのニーズヒアリング.....	100
5.6 ターゲットプロセッサの概要.....	102
5.7 統合実装実験と評価.....	105
5.7.1 統合時特有の動作確認.....	105
5.7.2 提案機械語モニタのプロセッサ統合実装結果.....	106
5.8 考 察	107
5.9 おわりに	108
第 6 章 結 論.....	109
6.1 本研究の成果	109
6.2 今後の展望と課題	111
謝 辞	113
参考文献.....	114
本研究に関する発表文献	121

図目次

図 1-1	FPGA の構造	...	6
図 1-2	CLB の構造	...	6
図 1-3	本論文の構成	...	9
図 2-1	故障率曲線 (バスタブカーブ)	...	12
図 2-2	IC の商品企画から出荷までの品質管理フロー	...	13
図 2-3	IC の熱抵抗と温度の関係	...	15
図 2-4	IC の熱モデル	...	15
図 2-5	FPGA 開発ツールの実装結果報告画面 (例 1)	...	18
図 2-6	消費電力関連情報の画面 (例 1)	...	18
図 2-7	FPGA 開発ツールの実装結果報告画面 (例 2)	...	19
図 2-8	消費電力関連情報の画面 (例 2)	...	19
図 2-9	高温試験に用いる恒温槽の外観	...	21
図 2-10	内部加熱器を含む FPGA の構造	...	21
図 3-1	マイクロプロセッサの消費電力の変化動向	...	23
図 3-2	温度測定システム外観と温度センサの位置関係	...	27
図 3-3	実験環境の構造	...	29
図 3-4	ターゲットプロセッサ (PU) のブロック図	...	32
図 3-5	スタンバイ状態における各温度センサの感知温度	...	35
図 3-6	スタンバイ状態におけるチップ温度	...	35
図 3-7	Fibonacci 数列を 100 MHz で計算した際の温度変化	...	36
図 3-8	動作クロック周波数変化のチップ温度 (T_{CB}) への影響	...	36
図 3-9	補正したチップ温度と動作周波数関係の HGP による差	...	39
図 3-10	1 ラインアセンブラ操作の様子	...	41
図 3-11	Fibonacci 数列計算時の温度と消費電力の変化	...	42
図 3-12	連続した加算で変化する bit 数	...	44
図 3-13	FPGA ボード写真 (表面)	...	45
図 3-14	FPGA ボード写真 (裏面)	...	45
図 4-1	リングオシレータ (RO) の回路例	...	48
図 4-2	ゲーテッド遅延素子 (GD) の回路	...	52
図 4-3	ゲーテッド遅延素子 (GD) で構成した提案 RO の構成	...	53
図 4-4	提案リングオシレータヒータ (ROH) の全体構成	...	54
図 4-5	加熱評価システムの構造	...	56
図 4-6	Vivado 開発環境の基本処理フロー	...	60
図 4-7	ROH 実装でカスタマイズした処理フロー	...	61
図 4-8	室温におけるチップ温度の測定結果	...	65
図 4-9	ROH モジュールの発振周波数分布	...	66
図 4-10	参照モジュールの発振周波数・遅延時間と遅延段数の関係	...	67
図 4-11	チップ温度 T_C と遅延段数の関係	...	68
図 4-12	チップ温度 T_C と GD 段数との関係	...	70
図 4-13	加熱温度変化と ROH モジュール数の関係	...	71

図 4-14	チップ温度の上昇率	...	72
図 4-15	ROHにより加熱されたFPGAデバイス付近のIR画像	...	73
図 4-16	外部加熱時のIR画像	...	73
図 4-17	簡易恒温槽の構成	...	74
図 4-18	簡易恒温槽の構造外観	...	74
図 4-19	FPGAデバイスのX線透視画像	...	74
図 4-20	FPGAデバイスのIR画像	...	74
図 4-21	複数のROHにより加熱されたFPGAデバイス付近のIR画像	...	75
図 4-22	熱制御機器(TC)の機能構成	...	78
図 4-23	熱制御機器(TC)により制御されたチップ温度と電源電流	...	80
図 4-24	熱制御機器(TC)により制御されたチップ温度のターゲット温度による差	...	80
図 5-1	従来MCMの実装フロー	...	84
図 5-2	提案iMCMの実装フロー	...	85
図 5-3	FPGA実装プロセッサの開発環境の構成	...	86
図 5-4	提案iMCMのブロック構成	...	87
図 5-5	実装したSMの機能モデル	...	90
図 5-6	基本ステート遷移図	...	90
図 5-7	プロセッサメモリのインタフェース	...	91
図 5-8	実験環境の画像	...	94
図 5-9	機能拡張したFPGAボードの画像	...	95
図 5-10	M(m) コマンドによるメモリ操作・表示の様子	...	95
図 5-11	iMCMの単体動作構成	...	96
図 5-12	プロセッサメモリサイズに対する実装時間とリソース占有率の関係	...	97
図 5-13	FPGAデバイス(XC7A100T)の構成要素数	...	97
図 5-14	SPARKLERのブロック図	...	103
図 5-15	ターゲットプロセッサを単一クロック実装した構造	...	104
図 5-16	プログラムトレースの様子	...	105
図 6-1	自律的に高温試験を行うFPGA	...	112

表目次

表 1-1	FPGA デバイスの進化	...	4
表 2-1	IC の温度区分例	...	12
表 2-2	IC 主要材料の熱関連特性	...	15
表 3-1	実験環境	...	26
表 3-2	実験に使用した FPGA デバイスの主な仕様	...	26
表 3-3	実験に使用した温度センサ	...	28
表 3-4	ターゲットプロセッサの ISA	...	30
表 3-5	プラットフォームモニタ (PM) が送信するデータ	...	33
表 3-6	温度測定システムの FPGA リソース占有率	...	33
表 3-7	Fibonacci 数列	...	37
表 3-8	Fibonacci 数列計算プログラムのトレース結果	...	41
表 3-9	チップ温度 (T_{CB}) と動作周波数関係近似の HGP による差	...	42
表 4-1	クロックシーケンサ (CS) の主な機能	...	57
表 4-2	加熱評価システムの各 FPGA リソース占有率	...	62
表 4-3	加熱評価システムの各 FPGA リソース占有率 (FPGA-m)	...	63
表 4-4	加熱評価システムの各 FPGA リソース占有率 (FPGA-s)	...	63
表 4-5	加熱評価システムの各 FPGA リソース占有率 (FPGA-l)	...	63
表 4-6	クロックシーケンサ (CS) に組み込んだ主なシーケンス	...	64
表 4-7	チップ温度 T_c と遅延段数との関係	...	68
表 4-8	チップ温度 T_c と GD 段数との関係	...	69
表 4-9	チップ温度上昇の時間率と ROH 群数の関係	...	72
表 4-10	提案 ROH と関連研究の緒元比較	...	76
表 5-1	モニタ機能と処理の関係	...	88
表 5-2	実験環境	...	92
表 5-3	実験に使用した FPGA プラットホームの機能	...	93
表 5-4	各 FPGA リソース使用数	...	98
表 5-5	モニタ機能ヒアリング者リスト	...	100
表 5-6	モニタ機能アンケートの判定方式	...	100
表 5-7	モニタ機能のヒアリング結果	...	101
表 5-8	SPARKLER の命令セットアーキテクチャ	...	102
表 5-9	FPGA リソース使用数に基づく回路規模	...	106

第1章 序論

1.1 本研究の背景

集積回路（以後 IC と呼ぶ）は、応用領域と利用者を拡大し、情報通信技術の原動力であり続けている。この勢いを支える製造プロセスの微細化・高集積化は、IC の高速化・高機能化などの性能向上、低消費電力化による持続可能な運用、及び低コスト化を進めてきた。現在では、7nm プロセスを用いる IC は 10 億オーダーのトランジスタを集積する。更に、2022 年下半期には 3nm プロセスのチップ量産が始まると伝えられている。

一方、最近の微細化・高集積化は、リーク電流増大や、プロセスマージン減少による物理的ばらつきの影響増大などの問題点も顕在化させた。物理的ばらつきの結果は遅延時間や消費電力に反映される [1], [2]。また、ウエハ大口径化にともなう少枚数のロット管理による生産性の低下など、製造プロセス世代毎に解決すべき課題もある。

近年、エンドユーザが身近で恩恵を受ける製品への IC 応用がますます進み、生活スタイルの変革を牽引している。代表的な例には、情報家電や車載機器、スマートフォンがある。これらの先進的な製品では、エンドユーザのニーズが多様であるとともに変化が速い。このため、頻繁な仕様変更や世代交代への対応が求められ、必然的に変品種変量生産の形態が多い。製造業全体では、インダストリー4.0 の中心にあるスマートファクトリの実現が期待される。その結果、IC にも変品種変量な供給が求められる。大規模な標準 IC の代表的な生産形態である少品種多量生産では、製品メーカーからの供給要求に応えきれない。そこで、その場でプログラム可能なゲートアレイ（Field Programmable Gate Array : FPGA）の応用が拡大している。

標準 IC の信頼性保証（reliability guarantee）は、初期不良品と製造マージンが少ない欠陥品を、出荷時に除去するスクリーニング（screening）で行われる。保証レベルに応じて、メーカーは IC に印加する電圧や温度の環境条件を設定する。また、スクリーニング実施には、実験室を超えた建屋規模の環境試験設備が必要になる。特に、チップ温度を 70°C 以上に加熱する高温試験には、大きな物理的・電気的コストを費やしている。

FPGA も厳しい動作環境における高信頼性が要求されてきた [3], [4]。FPGA が自動車、通信インフラ、及び製造ラインに代表される高信頼性製品に採用されるようになったためである。この要求に応えるため、メーカーは同一の FPGA を複数の速度等級と温

度範囲で区分する。FPGA 応用製品が要求する信頼性と、許容できるコストを同時に満足するような多くの選択肢が設定されている。これらの区分は、FPGA 出荷時のスクリーニング条件にしたがって選別される。

標準 IC とは異なり、購入後にユーザが回路実装する FPGA の信頼性保証方法は、信頼性試験、または設計保証で行われる。信頼性試験は、小規模なスクリーニングであるだけでなく、保証すべき信頼性を阻害する設計要因を分析する必要がある。環境試験設備とともに、IC テスタのような高コストの試験装置も必要になる。

設計保証は、信頼性試験に比してコスト面で優れている。設計値と実力値間のマージンを十分に確保することにより、回路同様に品質 (quality) を作り込む。設計保証の困難さとして、マージン設定を動作速度・集積度・消費電力に与える影響を最小にすることである。また、メーカーが提供するユーザ向け FPGA 開発環境には、設計保証に備えた消費電力や、FPGA チップ温度を見積る機能の組込みを進めている。しかしながら、実装する回路の種類・性質により、見積り精度のばらつき幅が桁違いとなる場合がある。

以上述べたように、ユーザが回路実装する FPGA の信頼性保証には、コストと保証レベルに関するトレードオフがある。

1.2 本研究の目的

1.1 に述べた観点から，本研究では物理的・電気的コストの低い FPGA 加熱方式の確立を目的として，FPGA に搭載されるプロセッサの高温試験を容易にする．提案する加熱方式の主機能は内部加熱であり，補助機能にチップ温度感知，及び温度制御がある．近年，チップ温度感知機能と，感知結果による動作をシャットダウンする保護機能を搭載する FPGA が増えている．本研究で把握する限り，内部加熱機能を搭載する FPGA は見受けられない．IC 全般を見回しても同様の状況である．

本研究は，FPGA の高温試験に資する能力を有する内部加熱機能に着目することにより，低コストで高信頼性保証レベルを確保する．内部加熱機能の最適化を中心に FPGA 加熱方式を開発し，高温試験の応用に貢献しようとするものである．結果として，継続的に信頼性保証できる安全・安心な FPGA 応用製品を提供する．

なお，本研究は FPGA チップの内部加熱による最大チップ温度，及び温度変化に対する FPGA の内部リソース比を，評価基準として設定する．

1.3 最新 FPGA の特徴

本研究の背景と目的を補足するため、FPGA の位置付け、技術動向、市場動向、及び内部構造について述べる。

FPGA の位置付け

ユーザがどこまで設計分担するかにより、IC は次のように分類されている [5]。 (1) 汎用 IC, (2) セミカスタム IC, (3) フルカスタム IC. 本研究では、(2)と(3)を一括してカスタム IC と呼ぶ。この分類から、FPGA は購入者として購入する時点では汎用 IC であるが、設計者としてプログラムした回路を FPGA 応用製品に搭載した時点ではカスタム品である。

現在では、IC 分類としての FPGA が浸透しているといえる。しかしながら、商用 IC として「プログラムに SRAM 方式を使った FPGA の原形」といわれる Xilinx 社の XC2000 ファミリは、1985 年 11 月の市場投入時には Logic Cell Array : LCA と呼ばれた。具体的な「Gate の Array (ゲート回路の配列)」に対して、抽象的な「Logic cell の Array (論理回路構成の配列)」と呼ぶことにより、ユーザに高位設計の可能性をアピールしたと考えられる。初期の FPGA はコンピュータやマイクロプロセッサの設計検証に活用されたが、製品へ搭載されることは少なかった [6]。XC2000 ファミリの XC2064 は 600~1,000 ゲート (2 入力 NAND ゲート換算) の回路プログラムを 2.0 μ m CMOS プロセスで製造した。表 1-1 に示すように、16nm プロセスで約 9 百万ゲートの回路プログラムを可能とする FPGA もある。

表 1-1 FPGA デバイスの進化

Series	LCA	Xilinx FPGA						
	XC2000	3	4	5	6	7	UltraScale	UltraScale+
Process (nm)	2,000	90	90	65	45	28	20	16
V _{CCINT} (V)	5.0 / 3.3	1.2	1.2	1.0	1.0	1.0	0.9	0.85
Year	1985	2004	2004	2006	2009	2011	2014	2016
Logic cell †	1,000	74,880	142,128	331,776	758,784	1,139,200	5,540,850	8,937,600
LUTs/Slice	(64CLBs)	2		4			8	
LUT	4-inputs	4 inputs		6 inputs				
I/O (max.) ††	58	633	896	1,200	1,200	1,100	1,404	1,976
Product family	XC2064		Virtex-4	Virtex-5	Virtex-6	Virtex-7 Kintex-7 Airtex-7	Virtex-US Kintex-US	Virtex-US+ Kintex-US+
High-End								
Mid-Range								
Low-Power Cost Effective								

† Logic cell = Slice \times 1.6 = gate counts in 2-input NAND gate. †† maximum specification in each Series.

FPGA の技術動向

規模の拡大ばかりでなく、製品に搭載できる FPGA が市場投入されてきた背景には ROM・RAM などのメモリ、PLL などのアナログ回路を混載する技術的進歩がある。更に、プロセッサコアや、独立した IC であった DSP が内蔵されるようになった [7]。近年では、ストレージや通信向けに高速インタフェースも内蔵されている。これらの混載や内蔵には、単純なロジックプロセスではなく、アナログ混載プロセスが使用される [8]。

更に近年では、メモリ混載プロセスの使用が必要になった。回路プログラムに用いる構成用メモリはほとんどが SRAM であり、電源の再投入毎に回路プログラム情報を外部から FPGA にダウンロードする必要がある。SRAM に代表される揮発性のメモリの信頼性に対して、ロジック回路よりも不安を感じるユーザがいる。そこで、構成用メモリに EPROM を用いるだけでなく、近年は組込み Flash メモリを搭載する FPGA がある。また、高い耐ノイズ性や放射線環境における動作保証も今後の課題である。最近では、メモリのように電荷で状態を記憶する代わりに、金属原子を小型スイッチとして再構成する技術の実用化が進められている [9], [10]。

FPGA の市場動向

FPGA を取り巻く製品の市場環境は、車載 (Advanced Driver-Assistance System : ADAS や自動運転)、AI における推論、インダストリー4.0, IoT を支えるクラウドコンピューティングやエッジコンピューティング、モバイルネットワークとアクセス、動画・画像や音楽・音声の低遅延伝送、などに広がりつつある [11], [12], [13]。これら市場の期待に応えるべく、ネットワークを介したデータのトラフィックは激増していく。それに伴い、第5世代移動通信システム (5th generation : 5G) から、6G へと大容量の基幹伝送装置や通信アクセス設備が必要になっている [14], [15]。

以上の高い技術要件に対応し、更に IC の開発コスト最小化とユーザへの供給時間最短化を実現するように、FPGA を使用しなければならない事例が増えつつある。ただし、市場拡大の機会を阻害するリスク要因に、最先端プロセスを使う IC 製造コスト低減と、さらに将来のプロセス技術の継続的進化がある。マイクロプロセッサやシステムオンチップで既に量産品に使用している 10nm 未満の最先端プロセスは、FPGA では数世

代ほど使用を遅らせている。近年のマイクロプロセッサメーカーによる FPGA 専門メーカーの大型買収の背景は、最先端プロセス利用の加速を狙ったもの、ともいわれている。

FPGA の内部構造

Xilinx 社 FPGA デバイスの代表的な構造を図 1-1 に示す [16]。論理ブロック (Configurable Logic Block : CLB) を X (横) 方向と Y (縦) 方向に CLB_{xy} を配列状に配置し、配線ブロック (Wiring block) が取り囲む構造は、多くの FPGA で共通である [17]。図 1-2 には CLB 周りを拡大した構造を示す。このデバイスの CLB は、6 入力 Look-Up Table : LUT を 4 組持つスライス (slice) 2 組を含む。同一 CLB 内 2 組のスライスは配線スイッチ (Switch matrix) を介して配線チャンネル (Wiring channel) に結合し、他の CLB と接続される。この構造により、同一 CLB 内における信号伝播速度の低下を抑えている。

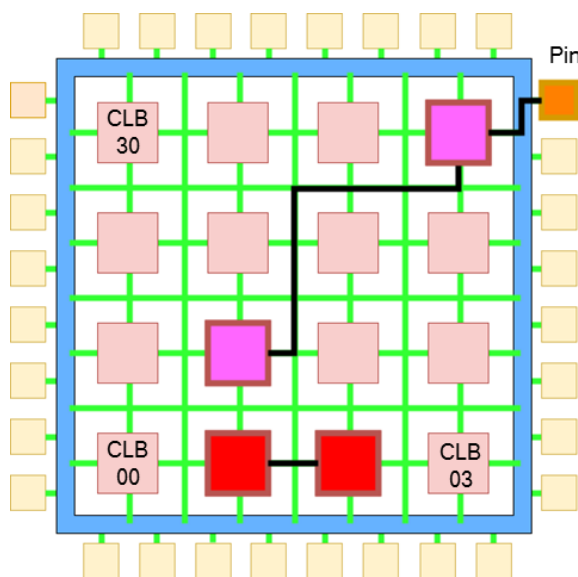


図 1-1 FPGA の構造

一つの回路が 2 組の CLB にまたがって配置される例

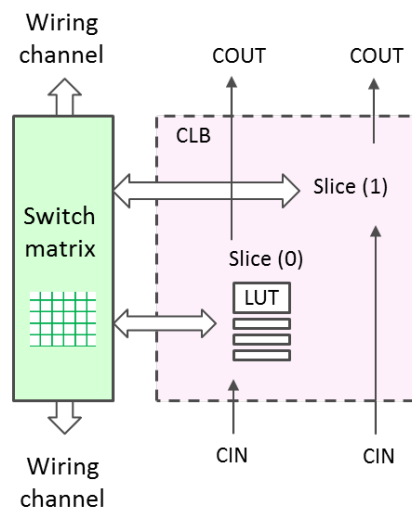


図 1-2 CLB の構造

CIN は各 Slice へのキャリー入力
COUT は各 Slice からのキャリー出力

1.4 本論文の構成

本論文の構成を図 1.3 に示す。本論文は 6 章から構成されている。以下各章ごとの概要を述べる。

第1章 「序論」

本研究の背景として FPGA に関する信頼性保証の特徴と解決すべき課題を示し、本研究の目的について述べる。加えて、最近の FPGA の特徴を述べることにより、背景と目的を補足する。更に以下に示すように、本論文の構成と各章の概要についてまとめている。

第2章 「FPGA における熱問題と熱利用」

熱が FPGA にもたらす不利益な問題と、対極にある有益な利用について述べる。問題については、まず IC 全般にわたる共通点から述べる。利用については、FPGA 高温試験向けの温度環境を供する内部加熱器について述べる。内部加熱器は、熱により上昇するチップ温度を要因として低下する IC 性能に着目し、FPGA に搭載するプロセッサを試験対象回路とする。

第3章 「FPGA 搭載プロセッサのプログラム実行によるチップ加熱の検討」

内部加熱器を実装する FPGA の規模的な制約を背景として、搭載プロセッサの加熱能力について述べる。プログラムとデータの性質がおよぼす消費電力への影響、及び加熱への影響からチップ温度への関係を把握し、この関係のモデル化検討について述べる。また、レジスタ転送レベルから新たに定義する小型プロセッサを FPGA に実装し、効率的にチップを加熱する発熱プログラムについて述べる。更に、4 種のプロセッサ動作速度と 5 種の発熱プログラムを組み合わせることにより、FPGA を加熱する条件について述べる。

第4章 「FPGA チップ温度制御用リングオシレータ加熱器の実装と評価」

FPGA の高温試験を必要とする状況を背景として、リングオシレータを利用し、FPGA デバイス自体が柔軟に発熱する内部加熱器について述べる。内部加熱器は最大定格が定める温度までのチップ加熱を目的とする。まず、内部加熱器をプロセッサと統合する構成について述べる。統合する構成を FPGA に実装し、内部加熱器を構成する遅

延素子の遅延段数と駆動段数, 加熱モジュール数の条件がチップ温度へ与える影響について述べる. 更に, プロセッサとともに実装した内部加熱器の特性について述べる. また, チップを推奨動作温度 85°C以上に加熱する条件と, 最大定格が定める 125°Cへの加熱可能性について述べる. 加えて, 最短時間で目標チップ温度へ加熱する温度制御回路を FPGA に実装し, 評価結果に基づいてプロセッサの高温テストへの応用可能性について述べる.

第5章 「FPGA 搭載プロセッサ向け機械語モニタのハードウェア実装」

高温環境下で回路の一部がマージン不足となる状態を背景として, 本来のモニタ機能を維持する実装手法について述べる. まず, ハードウェア記述言語で記述され, プロセッサとともに FPGA に実装する機械語モニタの実装手法について述べる. 提案する実装手法は, プロセッサの完全動作, 及びクロス環境の整備状況に依存しない. 更に, 機械語モニタの機能選択にあたって実施したプロセッサ設計経験者 6 名へのアンケートと, 回答結果について述べる. 加えて, 選択した機能とともに FPGA に実装したプロセッサについて述べる. このプロセッサは第3章および第4章において実装するプロセッサと同一である. 提案実装手法は, プロセッサが高温環境下でマージン不足となる一時的・部分的な故障状態にあっても, モニタ機能の提供を維持すると考えられる.

第6章 「結論」

本章では, 本研究の総括と今後の展開を述べる.

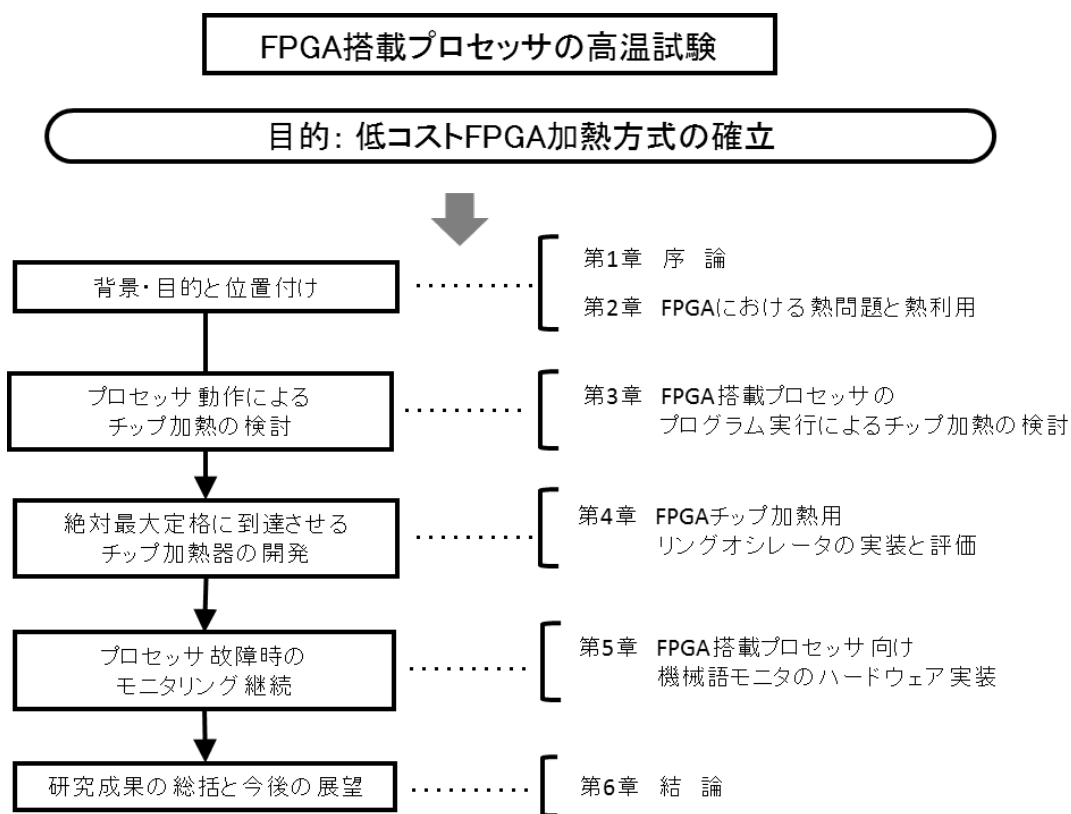


図 1-3 本論文の構成

第2章 FPGAにおける熱問題と熱利用

2.1 はじめに

本章では、熱が FPGA にもたらす不利益な問題と、対極にある有益な利用について述べる。問題については、まず IC 全般にわたる共通点から述べる。利用については、FPGA 高温試験向けの温度環境を供する内部加熱器について述べる。内部加熱器は、熱により上昇するチップ温度を要因として低下する IC 性能に着目し、FPGA に搭載するプロセッサを試験対象回路 (Circuit Under Test : CUT) とする。

以下、本章は次のように構成される。まず、2.2 で IC の熱問題について受動的側面と能動的側面からの影響を述べる。2.3 では発熱の要因と温度への影響について熱モデルを交えて述べる。つぎに、2.4 で FPGA 固有の熱問題について実例を交えて述べる。そして、2.5 で FPGA の内部加熱器の概要を説明する。最後に、2.6 で本章を締めくくる。

2.2 ICの熱問題

IC にとっての熱問題には受動的側面と能動的側面がある。受動的側面では、他から移動した熱により、IC の性能と数値表現が困難な特性が影響を受ける。一方、能動的側面では IC 自体が発熱体となり、IC 応用製品あるいは他の部品に影響を与える。受動的側面では、信頼性面で可逆性の劣化と、非可逆性の劣化あるいは破壊に至る影響がある [18]。また、温度上昇にしたがって (1) キャリア移動度の減少によるスイッチング性能の低下、(2) サブスレッショルド特性の悪化による静止電流の増加、(3) ラッチアップによる IC 破壊可能性の増大、(4) 配線抵抗の上昇による信号伝播速度の低下、(5) 熱伝導率の減少によるチップ温度上昇率の増加、等による特性への悪影響が知られている。反対に、液体窒素で得られる 100K (-173°C) 以下の低温における IC の動作により、低電力・高性能なコンピュータが発表され、注目された [19]。取扱いに大きな困難を伴わない室温レベルの動作にあっても、IC の性能・信頼性にとって、片や低温環境では利点が多く、片や高温環境では欠点が多い。

リーク電流の増大

上述の (2) について、プロセス技術進化の影響を補足しておく。100nm 以下のプロセスでは高速動作を狙うべく、しきい値電圧 (threshold voltage) を 2V より下げオン電流を増加させる際、ソースとドライン間の不要なサブスレッショルドリーク電流 (subthreshold leakage current) が増加する。サブスレッショルドリーク電流は温度変化に対して正の特性を持つ。この特性から、消費電力の増大によって温度が上昇するとサブスレッショルドリーク電流も増大するため、熱暴走に注意を払わねばならない。16nm プロセスを使用した FPGA では、1W オーダのリーク電流を想定しておく必要がある。更に、消費電力のピーク値抑制も重要であり、時分割に分散して回路を立ち上げることも有効と考えられる [20]。

信頼性に対する高温の影響

IC への高電圧や高温などのストレス印加は、IC の劣化を加速する。IC メーカーは市場への出荷時に、(1) 機械的、(2) 電氣的、(3) 熱的なストレスを与えるスクリーニングを行う。出荷後の故障発生は機械や装置で用いられる故障率曲線あるいはバスタブカーブ (bathtub curve) が、故障率の時間変化モデルとされている (図 2-1)

[21]. 代表的な故障率の単位には、時間当たりの故障件数 (Failure In Time : FIT) があり、一つの装置が 10^9 時間あたりに発生する故障件数を表す。故障率曲線は、(1) 初期故障期 (early failure period), (2) 偶発故障期 (random failure period), (3) 磨耗故障期 (wear-out failure period) で構成される。なお、IC の信頼性を考慮して 4 種類の温度関連仕様をメーカーが指定する。(1) 保存温度, (2) 絶対最大定格温度, (3) 推奨動作温度, (4) 室温。(1), (2) 及び (4) はメーカーと製品に依らずほぼ同じ (それぞれ 150°C , 125°C , 及び 25°C) である (表 2-1 (a)). (3) には IC 応用製品群別により温度区分があり、特に最大推奨動作温度の設定は細分化されている (表 2-1 (b)). 本研究では、すべての温度区分のうちで最大推奨動作温度を「高温」とする。表 2-1 (b) の例では 70°C である。

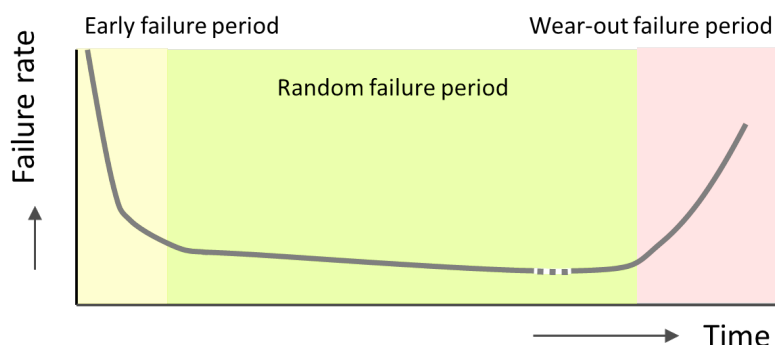


図 2-1 故障率曲線 (バスタブカーブ)

故障率 (Failure rate) は FIT または MTBF^{-1} で表される。

表2-1 ICの温度区分例

(a) 温度関連の主なメーカー指定仕様

温度に関わる仕様	記号	例 ($^{\circ}\text{C}$)
保存温度	T_{stg}	$-65 \leq T_a \leq 150$
絶対最大定格温度	$T_{\text{j(max)}}$	$T_{\text{j}} \leq 125$
推奨動作温度	T_{opr}	$-40 \leq T_{\text{j}} \leq \text{max.}^{\dagger}$ $0 \leq T_{\text{j}} \leq \text{max.}^{\ddagger}$
室温	T_a	25

(b) 温度区分別最大 T_{opr}

温度区分	max. ($^{\circ}\text{C}$)
自動車 (制御) †	125
自動車, 産業 †	110
拡張 †	100
民生 ‡	85
玩具 ‡	70

† maximum temperature of wide range operating temperature.

‡ maximum temperature of standard range operating temperature.

信頼性に対する低温の影響

以上述べたように、IC の低温における動作は、性能・信頼性にとって好都合な条件である。出荷時のスクリーニングにおいて、単なる低温環境はストレスとならない。低温における動作と高温における動作を繰り返す温度サイクル試験は、IC を構成する異なった材料の機械的ストレスによる信頼性低下を評価する。温度サイクル試験では、最低推奨温度への冷却及び最大推奨動作温度への加熱からなる温度サイクルを、100~1,000 回オーダで繰り返す [22]。サイクルの一例には、10 分間の低温維持→30 分の加熱移行時間→10 分間の高温維持→30 分の冷却移行時間、がある。

商品出荷までの高温試験

出荷可能な製品を検査するスクリーニングとは別に、品質管理フローには各種ストレスの印加により性能・品質・信頼性の評価工程がある。図 2-2 に示す品質フローでは、IC は出荷されるまでに合計 3 回、高温環境において評価及び検査される。製品設計の工程において、高温環境で設定される特性・品質・高温の目標を達成するように、設計上の適切な余裕を作り込むことが必要である。

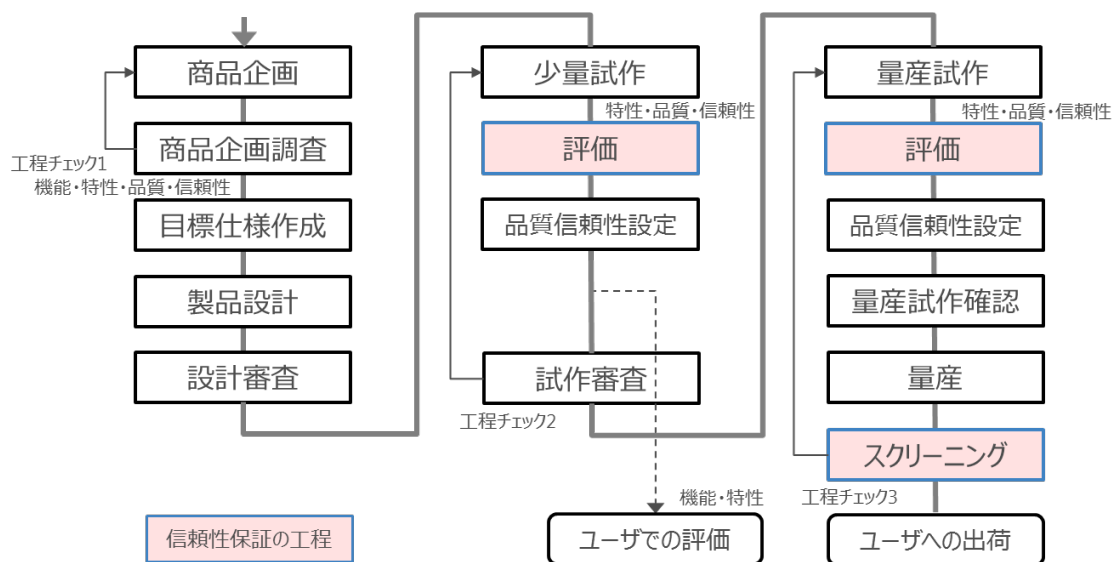


図 2-2 IC の商品企画から出荷までの品質管理フロー [5]

既存のプロセス技術・パッケージを使用する場合

2.3 ICの熱モデル

ICの発熱源はチップ上の回路と配線であり、式(2-1)に表すように動作電力 $P_{\text{operation}}$ とリーク電力 P_{leak} の和となる電力 P の消費で発熱する。 $P_{\text{operation}}$ は式(2-2)に表すように、クロック信号のような動作周波数あるいは動作頻度に依存する P_{dynamic} と、依存しない P_{static} で構成される。

$$P = P_{\text{operation}} + P_{\text{leak}} \quad (2-1)$$

$$P_{\text{operation}} = P_{\text{static}} + P_{\text{dynamic}} \quad (2-2)$$

発生した熱エネルギーは、チップからパッケージを介して、回路基板 (Printed Circuit Board : PCB) や周囲空気へと移動する。パッケージ上面からは対流で外気へ、パッケージ下面と端子からは伝導で移動し、移動量に応じて温度が変化する [23]。移動する熱エネルギーは材料により異なり、熱関連の特性を表 2-2 に示す [24]。

パッケージングされた IC が回路基板に搭載された状態で、チップで発生する熱と次に示す 3 種類の各境界における温度は、材料の熱抵抗 (thermal resistance) を使って表現される (図 2-3)。(1) チップ上の PN 接合部 (junction), (2) パッケージの表面 (case), (3) 回路基板 (board), (4) 周囲の外気 (ambient)。熱抵抗 θ (K/W) は、熱伝導率 λ (W/m·K) に物質の面積 A と厚さ m を反映させた式(2-3)で表わされる。

$$\theta = \frac{m}{\lambda \times A} \quad (2-3)$$

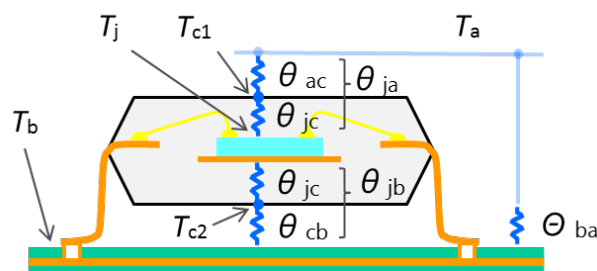
$\theta_{\text{ja}}, \theta_{\text{jb}}, \theta_{\text{jc}}$ は国際的業界団体である JEDEC で定められた条件下における実測値を基にメーカから提供される [25], [26], [27]。熱抵抗から各境界の温度計算例を式(2-4)に示す。PN 接合部 (以後ジャンクションと呼ぶ) - 周囲間で θ_{ja} の熱抵抗を持つ IC が、周囲温度 T_a で電力 W を消費した際のジャンクション温度 T_j を求めている。

$$T_j = T_a + \theta_{\text{ja}} W \quad (2-4)$$

図 2-4 は図 2-3 を簡易化するとともに、チップ上にある複数熱源 ($T_1 \sim T_m$) からの熱を、平衡してジャンクション温度 T_j とするモデルを示している。また、チップの T_j が定常的に 10°C 上がる毎にデバイスの寿命は約 1/2 になり、故障率は約 2 倍になるといわれる [28]。チップへの電源供給時に T_j を低く保持することが、高信頼性製品にとって原理原則である。

表2-2 IC主要材料の熱関連特性

ICの 主な材料	用途	熱伝導率 λ (W/m·K)	熱拡散率 a (m ² /s) $\times 10^{-6}$	熱膨張率 α (m/K) $\times 10^{-6}$
銀	ボンディング材	427	172	18.9
銅	チップ配線, リードフレーム	401	105	16.8
金	リードワイヤ	318	129	14.2
アルミニウム	チップ配線, ヒートシンク	238	97	23.6
シリコン	チップ	168	101	2.6
鉄	(42アロイの主材料)	81	21	11.7
はんだ	ボール (端子)	55	31	25
セラミクス	パッケージ	32	11	4.4
42アロイ	リードフレーム, ピン (端子)	15	4	4.2
ニッケル	(42アロイの副材料)	13	23	12.8
エポキシ樹脂	パッケージ, 回路基板	0.3	0.15	62
空気 @300K	周囲 (外気)	0.02	0.02	—



熱抵抗

- θ_{ba} : ボード - 周囲(外気)間
- θ_{jb} : ジャンクション - ボード間
- θ_{ja} : ジャンクション - 周囲間
- θ_{jc} : ジャンクション - ケース(裏面)間
- θ_{jc} : ジャンクション - ケース(表面)間

温度

- T_a : 周囲(外気)温度
- T_b : ボード温度
- T_{c1} : ケース(表面)温度
- T_{c2} : ケース(裏面)温度
- T_j : ジャンクション(チップ)温度

図 2-3 IC の熱抵抗と温度の関係

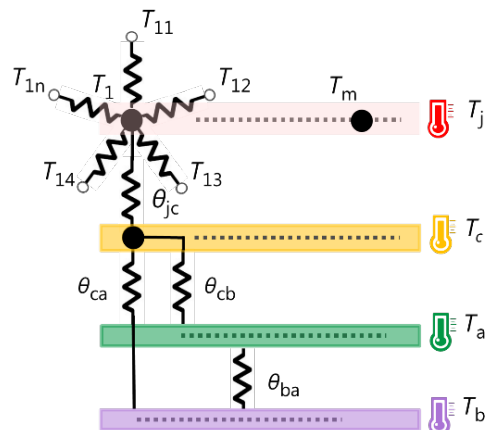


図 2-4 IC の熱モデル

2.4 FPGAの熱問題

ユーザがFPGAを使用するための熱と温度に関する情報として、メーカーは次の3点を定めている。(1) 絶対最大定格としての最大ジャンクション温度、(2) 温度区分で指定される推奨動作温度、(3) パッケージ関連の熱抵抗。一方、発生する熱の管理は、もっぱらユーザに委ねられている。このため、メーカーが提供するFPGA開発環境には、熱を見積る機能も含まれる。

図2-5に開発環境が報告する回路合成と実装結果の一例を示す。実装した回路は第4章で述べる加熱評価システムの一部であり、開発環境はXilinx社のVivadoツール(v2016.2)である[29]。この回路は32bitプロセッサコアとプロセッサメモリを含み、プログラムを実行できる。LUT占有率が82%となる設計である。図2-5には複数の情報表示窓があり、タイミング情報、リソース占有率、消費電力、及びチップ温度などが示されている。図2-6は消費電力とチップ温度の見積り結果を表わしている。図2-6(a)はおよそ560Wの消費電力と、125°Cのジャンクション温度を見積った結果を示している。過電流時に電源遮断する安全策を講じた上で、この回路をFPGAデバイス上で実際に動作させたところ、実測した消費電力は数W程度であった。

開発環境のうち電力や温度を見積るツールの多くは、外部からの入力値を内部回路に粗い粒度で扱う。更に、プロセッサのようにメモリに記憶されたストアドプログラム(stored program)により回路状態が変化するような場合は、見積値と実測値間の乖離が広がると考えられる。FPGAデバイスの破壊回避を目的として、最悪条件(worst case)における見積り精度を高めるために、ユーザが回路の使用率を設定する電力見積りツールも用意されている[30]。一方、回路への知識が不十分なプログラム開発者(プログラマを含む)にとっては、自動的に使用率が設定されるツールが望ましい。このように、プロセッサを搭載するFPGAの消費電力と、これに応じたチップ温度の見積りは困難である。

次に、以上に述べた見積値≧実測値とは逆の例を説明する。図2-6と同様に、図2-7は開発環境が報告する回路合成と実装結果の一例を示す。回路は8bitプロセッサコアを含み、LUT占有率は81%とFPGAリソースの面からも、図2-5で実装した回路に近い規模である。一方、プロセッサコアが異なる回路面の差に加えて、クロック信号に100MHzのタイミング制約(constrain)を使用した実装面の差異がある。図2-8(a)は28.1°Cのジャンクション温度を見積った結果を示している。回路記述による加熱器を実装したこの回路は、FPGAデバイスが自チップを100°Cに近いジャンクション温度に加

熱した。見積値≪実測値の結果となる実例を示した。

以上述べたように、消費電力とジャンクション温度の見積り精度が極端に小さい場合がある。想定を超えた極端な見積値≪実測値の関係は、FPGAデバイスだけでなくFPGA応用製品にとっても避けなければならない。このため、ジャンクション温度の危険領域への到達を検知すると、シャットダウンシーケンスを起動し、ICの破壊から保護する機能を内蔵するFPGAデバイスもある [31]。保護機能を内蔵するFPGAデバイスにとっても、構成する回路によっては実際の発熱と見積値の差を重視する必要がある。特に、ストアプログラム(参照処理)を実行するプロセッサを内蔵するような場合に、この傾向が強くなる問題を抱える。

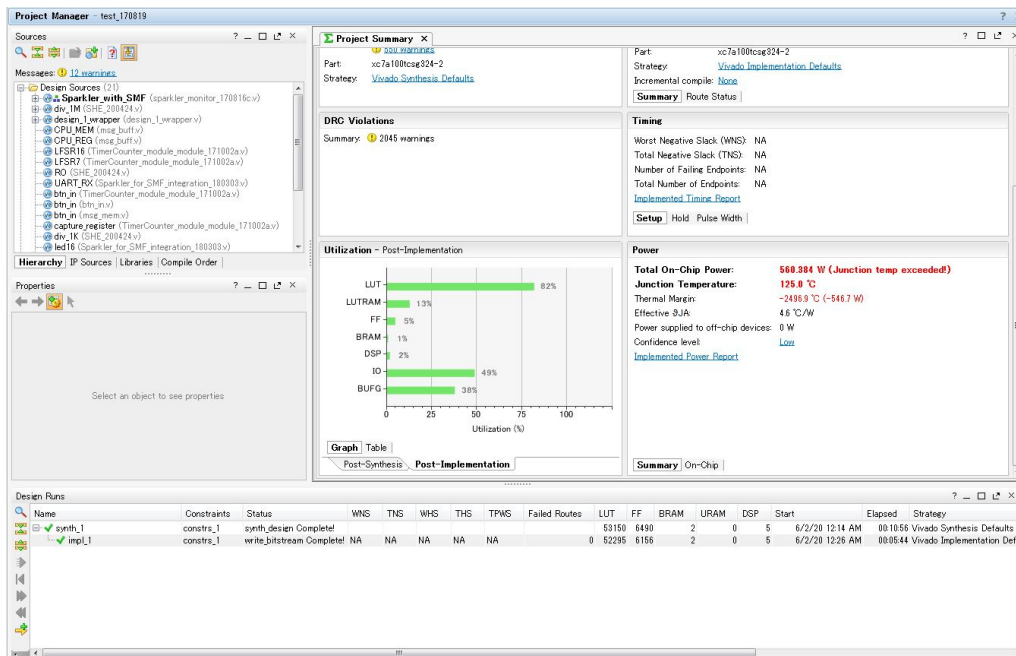
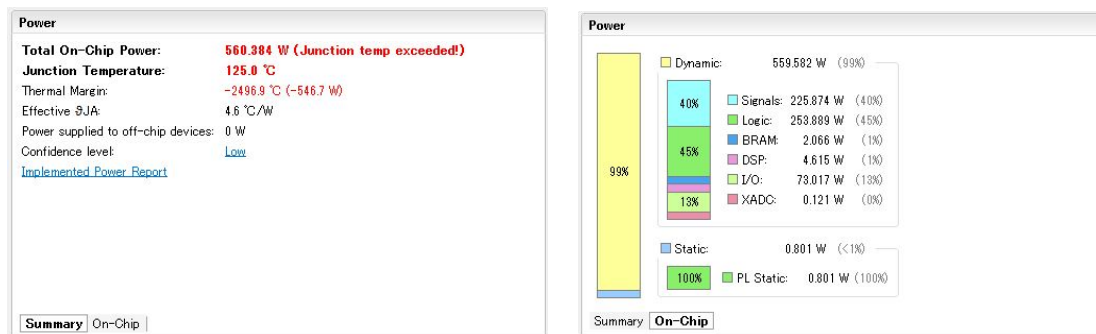


図 2-5 FPGA 開発ツールの実装結果報告画面 (例 1)



(a) 消費電力とチップ温度関係

(b) 詳細な消費電力

図 2-6 消費電力関連情報の画面 (例 1)

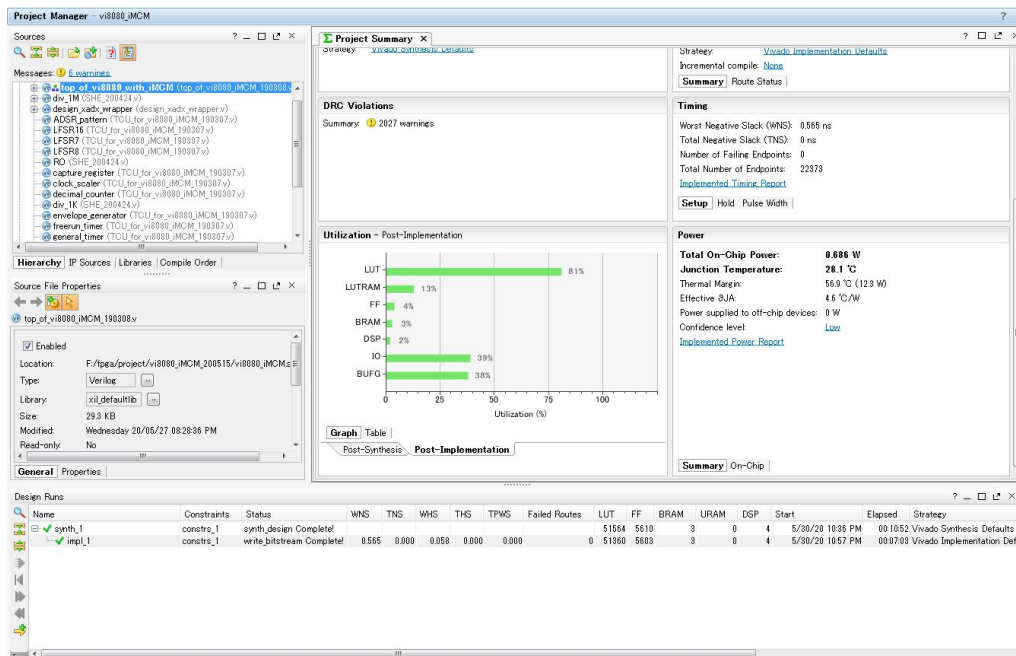
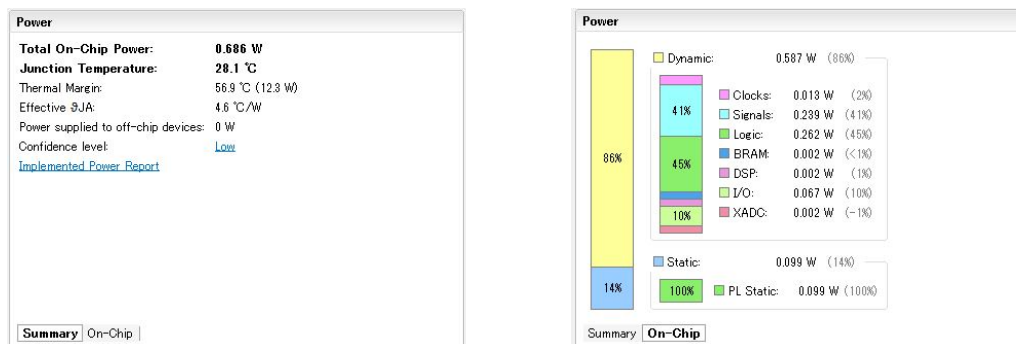


図 2-7 FPGA 開発ツールの実装結果報告画面 (例 2)



(a) 消費電力とチップ温度関係

(b) 詳細な消費電力

図 2-8 消費電力関連情報の画面 (例 2)

2.5 FPGAの熱利用

以上で述べたように、ICの発熱はプロセス進化に伴い増大する問題として扱われている。さらに、ICの故障発生は、動作時間によって故障モードと故障率の傾向が異なる。この傾向を捉えられれば、磨耗故障により寿命を迎える予兆を感知できる。予兆感知にはテスト回路内部の遅延時間を判定する回路を組み込む [32], [33], [34], [35], [36]。高温環境ではICの性能は低下する方向にあり、設計保証された動作マージンが削られ、故障の可能性が増す。そこで、あえてICを高温環境で動作させ、小さな動作マージンをあぶり出す。一例として、出荷時のICテストに高温BIST (Built-In Self-Test) がある。高温BISTはICが動作する動作環境と、ICを高温に加熱する試験環境を揃える必要がある。このような熱利用は、メーカーからのIC出荷時、あるいはユーザーからの製品出荷時に行われる。恒温槽 (temperature chamber) は高温試験に用いる環境試験設備である。100~1,000個オーダのICを同時に高温環境でテストするように100リットルオーダの内容量を持つ (図2-9)。

提案する内部加熱器は、製品の開発段階だけでなく製品の運用段階においても、FPGAデバイス内部からのチップ加熱を可能にする。内部加熱器 (internal heater) を含むFPGA全体の構造を図2-10に示す。図2-10には高温試験の対象CUTとなるプロセッサ (processor) を含む。1.3に述べたCLB及びWiring blockからなるFPGAの内部構造を、加熱に適用する。内部加熱器の加熱特性として、(1)安全性がまず必須であるが、加えて(2)効率性、(3)安定性、などが期待される。少ないFPGAリソースで発熱させる方式にはCLBの出力を内部電源 (V_{CCINT}) あるいはグラウンド (ground: GND) への短絡 (short) が最も効果的である。一方で、この方式は出力トランジスタの破壊や致命的な劣化を排除できず、安全性を求められる内部発熱器には選択肢とならない。図2-10に示すように、温度センサ (temperature sensor) と温度制御器 (temperature controller) によるFPGAチップ上の付加機能により、内部発熱器の加熱を制御する。これらの付加機能には安価な実装が期待される。FPGAはアナログ回路やメモリ回路などが不要な限り、論理回路の再構成によりこの期待に応えられる。内部加熱器からの発熱だけでなく、プロセッサからの発熱も熱結合 (thermal connection) により温度感知される。

上述のように、内部発熱器は搭載プロセッサをCUTとして、高温試験の結果から動作マージンを測定することにより、開発段階及び運用段階においてFPGAが製品にと

って信頼し得る (dependable) 部品か否かの判定に応用できる。



(a) 全体外観

外寸 : 910×1,590×1,073 (W×H×D mm)

(b) 正面外観

内容量 : 225 リットル

図 2-9 高温試験に用いる恒温槽の外観

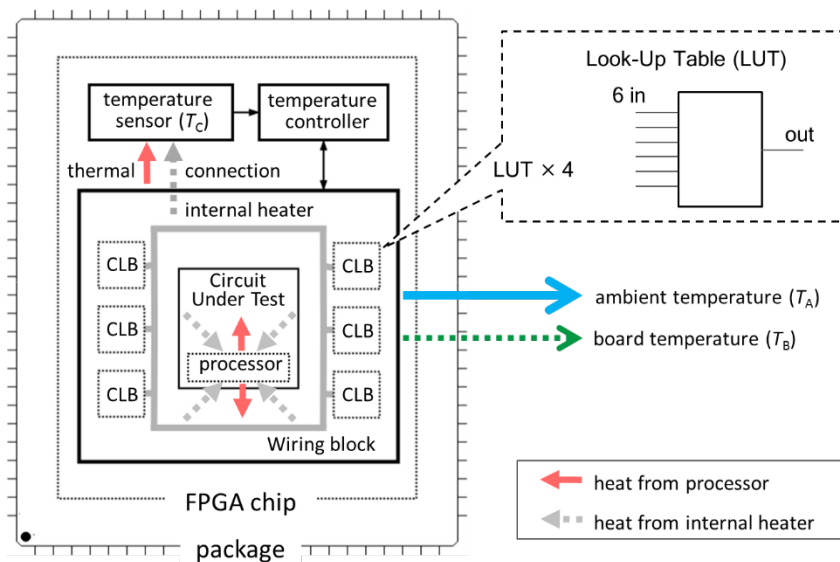


図 2-10 内部加熱器を含む FPGA の構造

多くの FPGA には temperature sensor がチップ上に標準搭載される

2.6 おわりに

本章では、熱が FPGA にもたらす不利益な問題と、対極にある有益な利用について述べた。問題については、まず IC 全般にわたる共通点から述べた。利用については、FPGA 高温試験向けの温度環境を供する内部加熱器について述べた。

内部加熱器は、熱により上昇するチップ温度を要因として低下する IC 性能に着目し、FPGA に搭載するプロセッサを試験対象回路 (Circuit Under Test : CUT) として、具体的な FPGA の構造について述べた。内部加熱器の使用で、恒温槽を使用しない高温試験が期待できる。

第3章 FPGA 搭載プロセッサのプログラム実行によるチップ加熱の検討

3.1 はじめに

プロセッサの進化を妨げる要因の一つに IC の消費電力増加がある。消費電力による IC 内部からの発熱は信頼性低下の問題を引き起こし、IC の寿命を短縮する。更に、この問題を回避するためのパッケージやシステム全体の冷却コストを上昇させてきた。図 3-1 に 1978 年に市場投入された Intel 社の 16bit マイクロプロセッサ i8086 から、近年までに市場投入された 16bit 世代以降のマイクロプロセッサの消費電力を示す。消費電力が 100 W を超えるプロセッサでは、消費電力が変換される発熱の問題（以後、熱問題と呼ぶ）は最も深刻である。高性能を要求されるサーバやワークステーション、及び高密度に集積されたクラウドコンピュータやマイニングマシンでは、安定した冷却機構の実装が今後も深刻な課題であり続けるだろう。また、スマートフォンにおけるエンドユーザの満足感は、アプリケーションの高度化とバッテリーでの動作の長時間化が両立することにより高まるため、10 W を超えるプロセッサの熱問題への取組みが加速している。

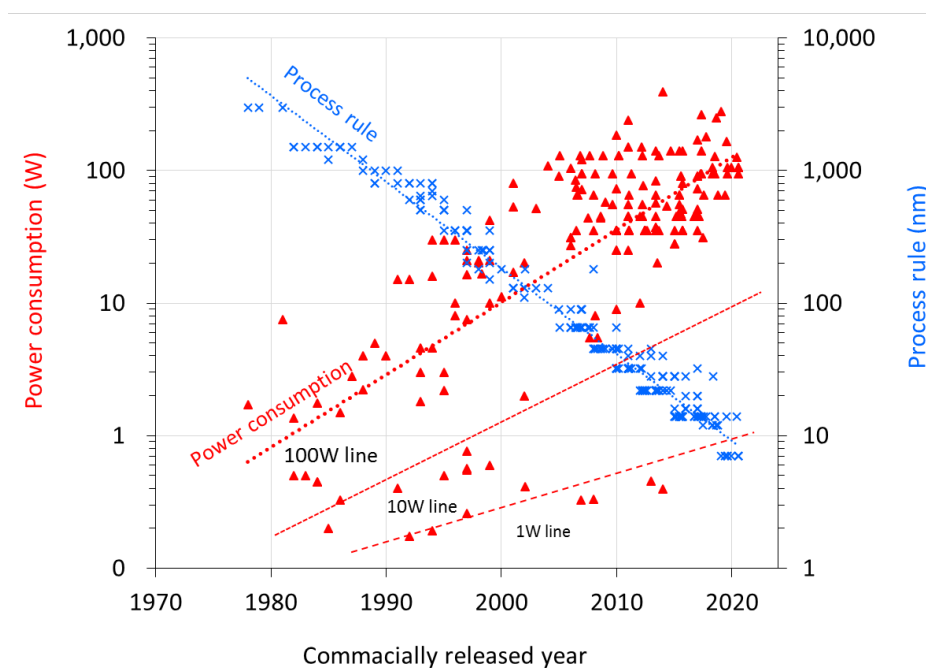


図 3-1 マイクロプロセッサの消費電力の変化動向
(データはメーカーからの発表値を使用した)

一方, 1 W を下回るサブワット (subwatt) での動作しか許容されない組込みシステムでは, 消費電力の正確な測定に際して小型実装と低コスト実現の課題がある. 組込みシステムの消費電力を測定する方法の一つとして, 電力供給線の途中に挿入したシャント抵抗 (shunt resistor) を電流センサとして使用する間接法 (indirect method) が知られている. この方法では, アナログ - デジタル変換器 (Analog-Digital Converter : ADC) が必要になる. このため, 組込みシステムに要求される速度, 精度, コストのいずれかを圧迫する. サブワットに分類される組込み機器では, センサや ADC を使用せずに消費電力を測定する方法が期待されている.

本章では, 組込みシステム用のボード上にプロセッサを回路実装する FPGA を想定して, プログラムとデータの性質がプロセッサ消費電力だけではなく, 消費電力による FPGA デバイスの加熱が影響するチップ温度への関係を検討する [37], [38].

以下, 本章は次のように構成される. まず, 3.2 でプログラムとデータの発熱への影響を議論する. 3.3 では, 予備実験の結果と得られた知見について議論する. 3.4 では, 実装実験装置を紹介する. 3.5 では, 発熱プログラム実行による発熱実験結果を議論する. 3.6 では, 参照発熱プログラムの発熱実験結果と消費電力測定結果からの関係を議論し, 3.7 で本章を締めくくる.

3.2 サブワット級 IC の熱問題

近年、性能重視のシステムではプロセッサの熱管理のために高レベルで多岐に渡る対策が講じられている [39], [40]. 消費電力が 10 W を超えるプロセッサを想定した代表的な研究には、急激かつ局所的な発熱の変化を複数のオンチップ温度センサによる感知 [41], 粒度の粗い温度感知をチップ全体に拡大するモデリング化 [42], 実行するソフトウェアの熱的振舞モデルの高精度化 [43], 発熱量の異なる複数のプロセッサを感知した温度による切替え [44]が含まれる. しかしながら小型で低消費電力動作が求められる組み込みシステムではシステムレベルの熱的挙動との関係が強くなり、これまでの関連研究の直接的な貢献に限られている場合がある. サブワット級 IC の温度は自体内の発熱だけでなく、他 IC の発熱を温度感知に反映させてしまう可能性が高くなる.

本章では、再構成可能な IC のひとつである FPGA にプロセッサを搭載するシステムを対象とし、オンチップ温度センサからプロセッサの消費電力を算出し、加熱されたチップ温度との関係を検討することを目的とする. そこで、検討結果の有効性を確認するために用意したハードウェアとソフトウェアの目的を明確にしておく. ハードウェアとしては電力最小化を目指してプロセッサの命令セットアーキテクチャ (Instruction Set Architecture : ISA) とマイクロアーキテクチャ (microarchitecture) を選択した. 一方でソフトウェアとしては電力最大化を目指してプログラムとデータを選択した. 発熱低減の条件を理解するためには電力消費を大きくするプログラム構造 (Hot program) やデータパターン (Hot data) の把握が早道と考えたためである.

3.3 チップ加熱実験の概要

3.3.1 プラットホームの選択

表 3-1 に本章で使用した実験環境一式を示す。実験環境の中核となる FPGA ボードには、低消費電力で動作する FPGA (Xilinx Artix-7 ファミリ) を搭載した開発評価システム一式 (以後、プラットフォームと呼ぶ) を選択した。この FPGA にはオンチップ温度センサがチップ上に組み込まれている。表 3-2 に搭載した FPGA の主な仕様を示す。この FPGA が搭載するリソースは、Artix-7 ファミリのなかで中規模である [45]。このため、本章の実験結果から小規模システムまたは大規模システムいずれへの実験の展開も容易と考えられる [46]。

表3-1 実験環境

ワークステーション (PC)	CPU: Xeon 3.5 GHz, メモリ: 16 GB
OS	Windows 7 professional (64bit)
FPGA 開発ツール	Vivado 2016.2 (Xilinx 社)
HDL	Verilog HDL
端末ソフトウェア	RLogin (Culti)
ターゲット FPGA デバイス	Artix-7 (Xilinx 社)
FPGA プラットホーム	Nexys4 DDR (Digilent 社)
デジタルマルチメータ	34465A (Keysight Technologies 社)
IR カメラ	FLIR ONE (FLIR 社)

表3-2 実験に使用したFPGAデバイスの主な仕様

項目	記号	仕様
デバイス名		Artix-7 (XC7A100T-1CSG324C)
プロセス (nm)		28
ロジックセル数		101,400
LUT数		63,400
パッケージ (mm)		324 ball chip-scale BGA 15×15×1.5
I/O数		210
熱抵抗 (°C/W)	θ_{ja}	18.2
	θ_{jc}	5.7
	θ_{jb}	3.25
推奨動作ジャンクション温度 (°C)	T_j	0~85
最大ジャンクション温度 (°C)	$T_{j(max)}$	125
推奨動作電圧 (V)	V_{CCINT}	0.95~1.05

3.3.2 温度センサの選択

FPGA へのプロセッサ実装の準備段階として、オンチップ温度センサが FPGA 自体の消費電力をどのように反映するのかを確認した。オンチップ温度センサの感知データから、電源投入後の回路プログラミングや回路の起動・静止・動作の状態に関わらずチップ温度の安定には 10~15 分を要した。

この確認結果から、ボード上とボード外部の自由空間に温度センサを追加し、そしてボード温度と周囲温度も同時に測定した。実験では、以下のように 3 つの温度センサを用意した。(1) オンチップ温度センサ (CTS), (2) 補正用ボード温度センサ (BTS), (3) 補正用周囲温度センサ (ATS)。表 3-3 に使用した 3 つの温度センサを示す。図 3-2 に FPGA ボード上の CTS と BTS, 及びボードに隣接した ATS の位置を示す。

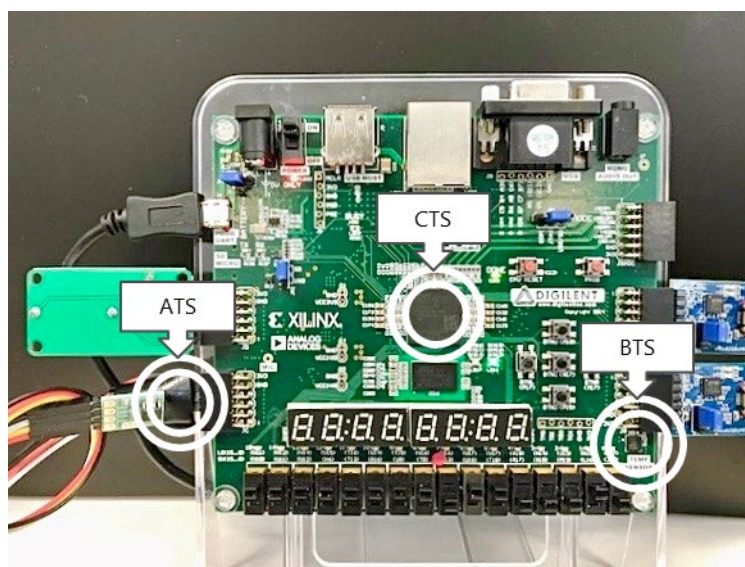


図 3-2 温度測定システム外観と温度センサの位置関係

表3-3 実験に使用した温度センサ

TS: Temperature sensor	for	Main specification	Device
Ambient TS (ATS)	T_A	0.078°C/LSB, $\pm 0.5^\circ\text{C}$	ADT7410 †
Board TS (BTS)	T_B	0.078°C/LSB, $\pm 0.25^\circ\text{C}$	ADT7420 †
on-Chip TS (CTS)	T_c	0.123°C/LSB, $\pm 4^\circ\text{C}$	XADC ††

† Analog Devices, Inc. product. †† Xilinx Analog-to-Digital Converter.

3.4 実験装置の概要とターゲットマイクロプロセッサの実装

図 3-3 に FPGA を中核とする実験環境の構造を示す。FPGA には、(1) 熱を発生する 32bit のターゲットプロセッサ (Processor : PU) と関連ブロック、(2) 発熱状況を間接的に温度で感知する温度測定システムを構成するブロックを、Verilog HDL 記述を用いて回路実装した。また、消費電力の簡易測定用に FPGA ボードの電源 (+Vpower と -Vpower の間) に直列にシャント抵抗 (R_{shunt}) を挿入した。

次に、3.2 で述べたように、PU の ISA とマイクロアーキテクチャは電力最小化を目指す。また、ともに変更が容易であるように回路実装はレジスタ転送レベル (Register Transfer Level : RTL) から設計した。以下 PU の設計に関連して、3.4.1 に ISA の選択肢について、また 3.4.2 にマイクロアーキテクチャの選択肢について説明する。

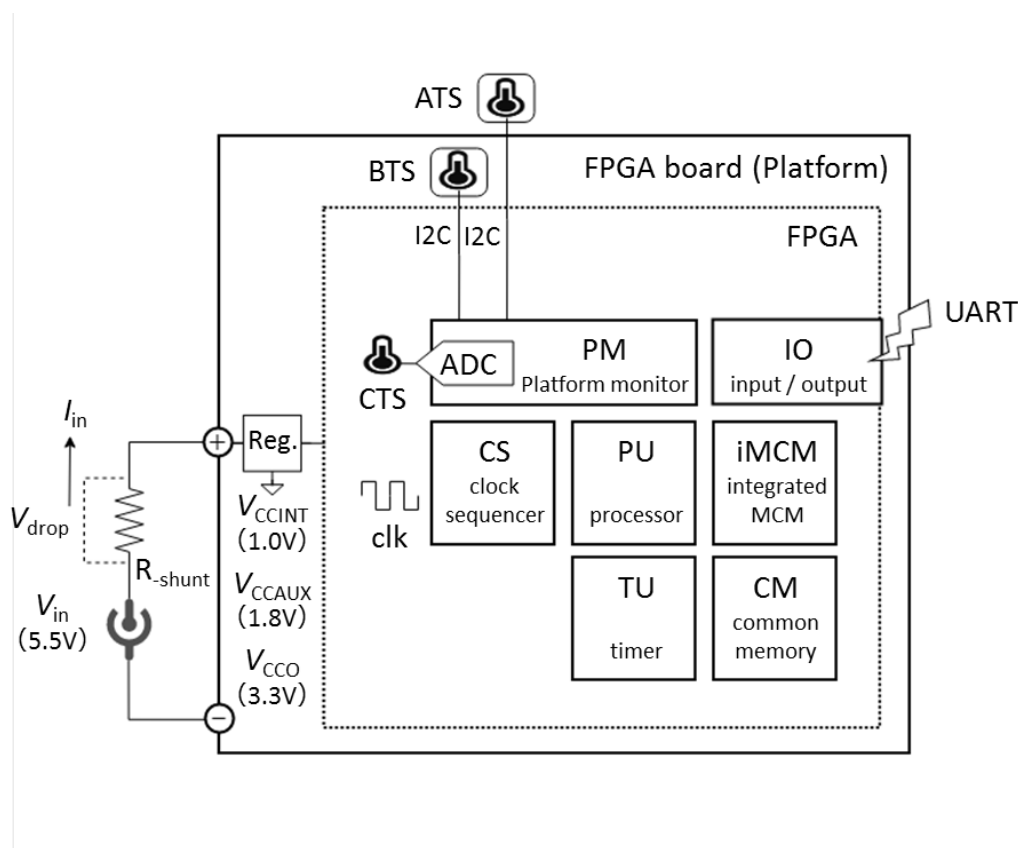


図 3-3 実験環境の構造

3.4.1 ターゲットプロセッサの命令セットアーキテクチャ

PUは表3-4に示すようにISAとしてRISC-Vのうち最小となる16bit命令長の圧縮型命令セット拡張(“C” Standard Extension)を採用している [47], [48].

表3-4 ターゲットプロセッサのISA

基本 ISA	RISC-V	
分類	RV32C	16bit 命令長, 32bit ワードデータ長
レジスタ		32 本 (32bit)
命令空間	命令 :	2^{30} ワード (32bit)
	データ :	2^{30} ワード (32bit)
命令†	演算 :	ADD, SUB, AND, OR, XOR, SLLI, SRLI, SRAI, ADDI, ADDI16SP, ADDI4SPN, ANDI
	ロード/ストア :	LW, LWSP, SW, SWSP
	分岐/ジャンプ :	J, JR, JAL, JALR, BEQZ, BNEZ, EBREAK
	転送 :	LI, LUI, MV, NOP

† ニモニク表現 <ADD> は <c.add> を置き換えた例

3.4.2 ターゲットプロセッサのマイクロアーキテクチャ

MIPS プロセッサに代表される現代のマイクロプロセッサでは5段階程度のパイプラインステージを有するマイクロアーキテクチャ, すなわちパイプライン処理方式の採用が多い [49]. この実装方式は命令処理能力のスループット (throughput) とレイテンシ (latency) のバランスがよい ISA と組み合わせて, はじめて期待される性能が提供できる. 一方, パイプラインステージを動作クロックに同期化するハードウェアが必要であり, この同期化ハードウェア固有の消費電力が性能とのトレードオフとなる.

PU には単一サイクルのマイクロアーキテクチャを採用した. パイプライン処理方式固有の消費電力を排除し, ソフトウェアの処理フローと処理データの消費電力への影響を浮き出させる. 図 3-4 に PU のブロック図を示す. プログラムカウンタ (PC) が指定する次のアドレス (next. Adr.) を動作クロック (clk) で同期化した後は命令メモリ (IM), 命令デコーダ (ID), レジスタファイル (RF), 命令実行ユニット (EX), データメモリ (DM) における命令処理がこれらのブロックを進む. 1 命令に必要な全処理を 1 クロック期間内で完了させる. このマイクロアーキテクチャは組込みシステム向けプロセッサとしての有益性と実現性を有していると考えられる. 1 クロック期間で命令処理を完了しなければならない回路段数が多いため処理サイクル時間が長くなるものの, 分岐ハザードやレジスタハザードを含む例外的な処理用のハードウェア構造を単純化・小規模化する. これらの変化により, 動作クロック同期動作の削除を超える消費電力削減が期待される. 更に, IM, DM に独立したメモリ空間を割り当てるハーバード・アーキテクチャ (Harvard architecture) を採用し, メモリハザードの発生を排して性能低下の抑制を図っている.

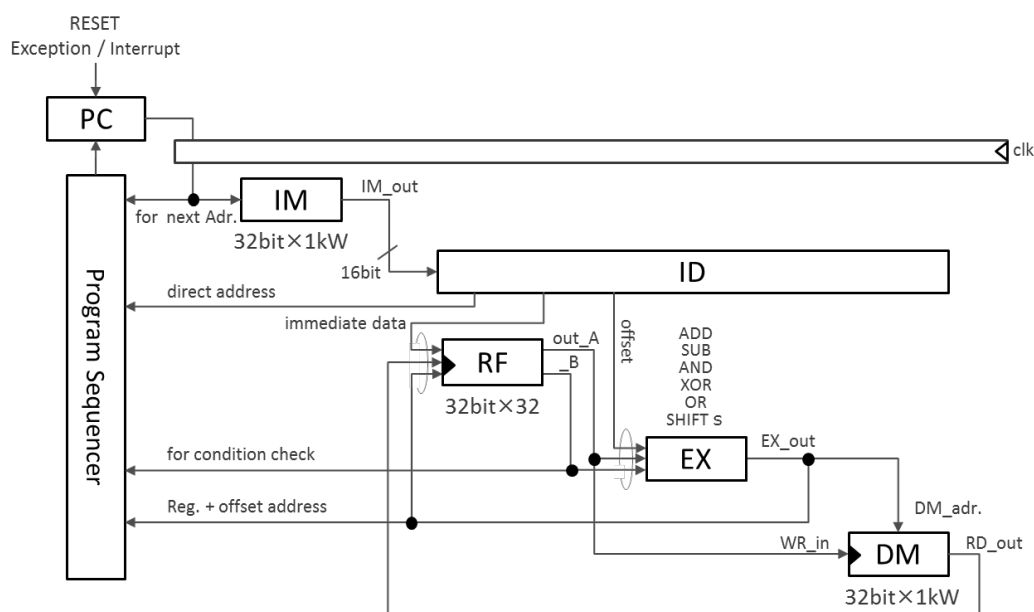


図 3-4 ターゲットプロセッサ (PU) のブロック図

3.4.3 FPGA への実装

FPGA には PU の他に次のブロックを実装した。(1) 内部メモリやレジスタを操作する統合機械語モニタ (iMCM), (2) 温度データを収集してホスト PC に送信するプラットフォームホームモニタ (PM), (3) プロセッサ動作のクロック周波数をあらかじめ設定された順序で供給するクロックシーケンサ (CS), 及び (4) 測定システム内のタイミング信号を発生するタイマ (TU), (5) LED 表示, UART 通信など FPGA 以外の IC を制御する周辺 IO (IO) で構成されている。iMCM は開発ツールが未整備な状態でプログラムの実行とデバッグを容易にする。PM はリアルタイムクロック (Real Time Clock : RTC) のタイミングに同期して, 3.3.3 で述べた 3 つの温度センサの温度データを RTC の計時タイミングに同期して測定する。測定されたデータは文字コードに変換され, UART インタフェースを介してホスト PC の端末ソフトウェアに送信され, PC に記録される。表 3-5 に PC に送信されたデータの一例を示す。左から (1) RTC の時間情報, (2) CTS のチップ温度データ, (3) BTS のボード温度データ, (4) ATS の周囲温度データ, (5) CS 内のシーケンス情報である。また, 表 3-6 に実装した FPGA の内部リソース使用数を示す。

表3-5 プラットホームモニタ (PM) が送信するデータ

	RTC	CTS	BTS	ATS	CS
1	13:59:59	37.3	28.8	26.4	(0-B-F)
2	14:00:00	36.9	28.8	26.4	(0-B-F)
3	14:00:01	37.3	28.8	26.4	(0-B-F)
4	14:00:02	37.0	28.8	26.4	(0-B-F)
5	14:00:03	36.8	28.8	26.3	(0-B-F)
6	14:00:04	36.6	28.8	26.4	(0-B-F)
7	14:00:05	37.3	28.8	26.4	(0-B-F)
8	14:00:06	37.0	28.8	26.4	(0-B-F)
9	14:00:07	36.5	28.8	26.4	(0-B-F)
10	14:00:08	37.4	28.8	26.4	(0-B-F)
11	14:00:09	36.8	28.8	26.4	(0-B-F)
12	14:00:10	36.8	28.8	26.4	(0-B-F)

表3-6 温度測定システムのFPGAリソース占有率

Block	LUTs	Utilized Ratio (%) [†]
PU: Processor Unit	5,979	9.4
CM: Common Memory	545	0.9
iMCM: integrated MCM	2,394	3.8
CS: Clock Sequencer	116	0.2
PM: Platform Monitor [†]	3,119	4.9
Total: 63,400 (all)	12,153	19.2

[†] including TU and IO.

3.5 発熱プログラムを用いた評価

3.5.1 予備実験の結果とフィードバック

実験の第一段階の予備実験として、対象システムの各温度の基本特性を確認した。実験は空調された室内で行ったが、室温は外気温と在室人数により $\pm 2^{\circ}\text{C}$ ほどの範囲で変化した。図 3-5 (a), (b), (c), (d) に春夏秋冬の無作為に選択した日の 0 時から 24 時までの 24 時間連続の温度データを示す。これら各図の下 3 データは、1 秒ごとに取得した ATS, BTS, CTS からの温度変化をそれぞれ示している。すなわち、(1) 実験環境の周囲温度 (T_A)、(2) ボード温度 (T_B)、(3) チップ温度 (T_C) の温度変化がわかる。 T_C のみ時間軸・温度軸を拡大して図 3-6 に示す。温度で $\pm 0.5^{\circ}\text{C}$ ほど、分解能 $0.123^{\circ}\text{C}/\text{LSB}$ のアナログ-デジタル変換であるため、 $\pm 3\text{LSB}$ ほどのノイズ的な変動がわかる。また、図 3-5 (a), (b), (c), (d) それぞれの最上段のデータは、60 項の単純移動平均 $T_C - T_B$ を示している。移動平均によって変動幅を約 10 分の 1 にまで縮小した。移動平均前の標準偏差 σ は 0.31 で移動平均後は 0.069 となり 0.1°C の感知データ差が 2σ 内に近くなる。この確認結果から、消費電力の見積もりにボード温度により補正したチップ温度 T_{CB} ($T_C - T_B$) 適用の妥当性が示唆される。

予備実験の結果からのフィードバックとして、次の条件の下で関連データを測定した。CTS がプロセッサの消費電力をどのように反映しているかを確認した。CTS の検知データからのフィードバックとして、以下 3 項目の実験条件を設定した。(1) 10 秒以上継続する、(2) 5 回測定した値の平均値を測定結果の集合とする、(3) FPGA に回路を書き込んでから 15 分以上のインターバル時間を確保する。

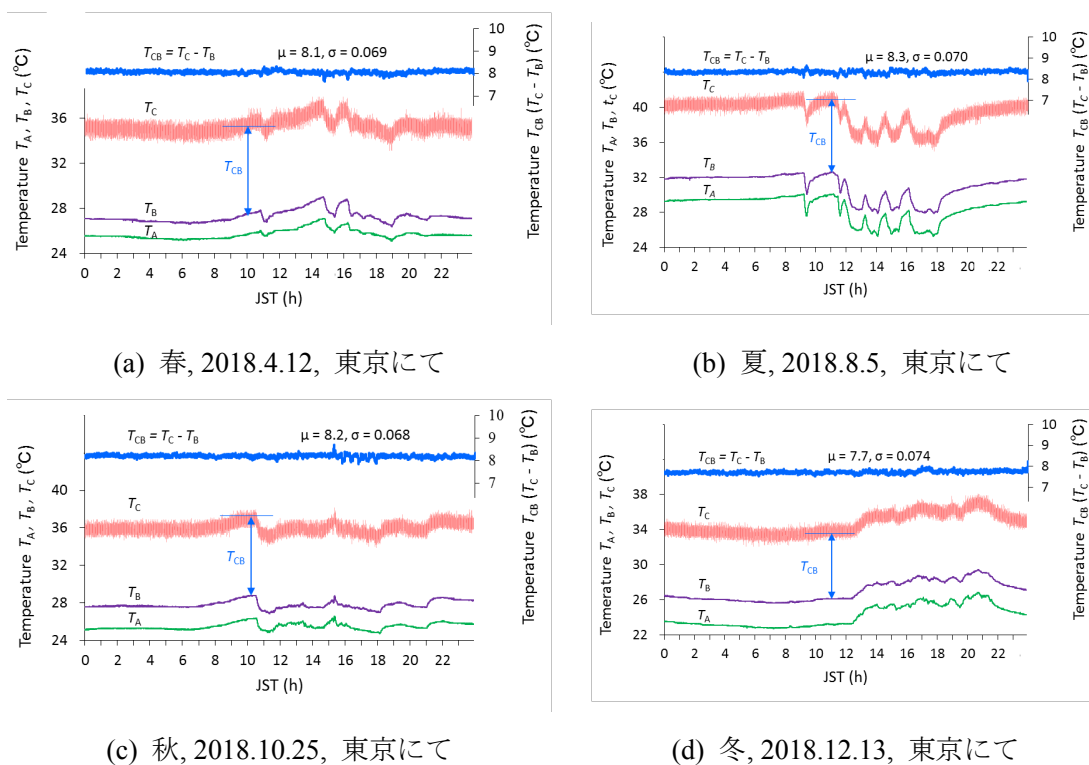


図 3.5 スタンバイ状態における各温度センサの感知温度

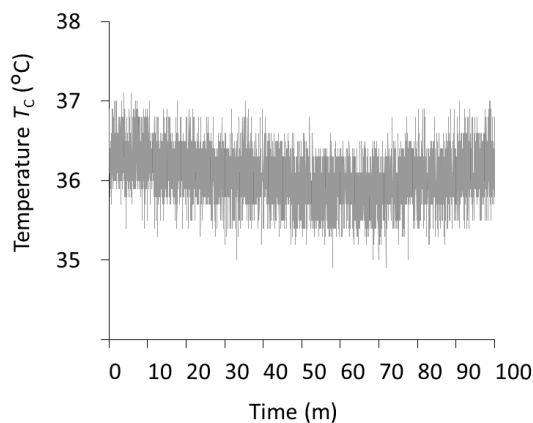


図 3-6 スタンバイ状態におけるチップ温度

3.5.2 プログラム実行による発熱の基本的な挙動

図 3-7 は PU でフィボナッチ数列 (Fibonacci sequence) の計算を 100 MHz クロックで 300 秒間 (300 億命令) 実行した際の T_{CB} を示す. 実行開始と実行終了時に緩やかな傾きの上昇/下降 (+slope / -slope) がある. この傾きはチップからパッケージへの放熱効果を示していると考えられる. 図 3-8 は最初の 100 秒は 20 MHz, 次の 100 秒は 50 MHz, 最後の 100 秒は 100 MHz の動作クロックに切り替えてフィボナッチ数列計算を実行した際の T_{CB} を示す. ここでもクロック周波数の切替え毎に T_{CB} に小規模で緩やかな傾きがある. 示された傾きは T_{CB} の上昇幅に依存していると考えられる. なお, 上述の図 3-7, 図 3-8 とともに 3.5.1 で実験条件として定めたとおり, 5 組の測定結果の平均を示している.

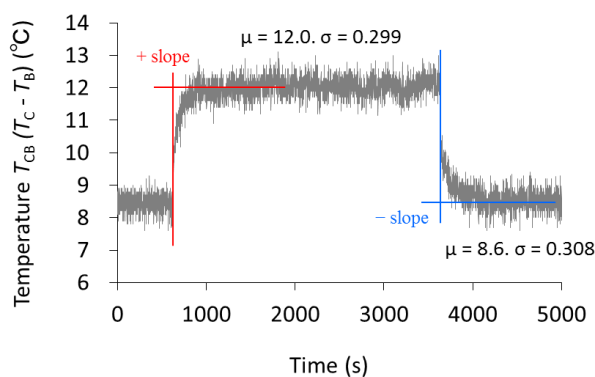


図 3-7 Fibonacci 数列を 100 MHz で計算した際の温度変化

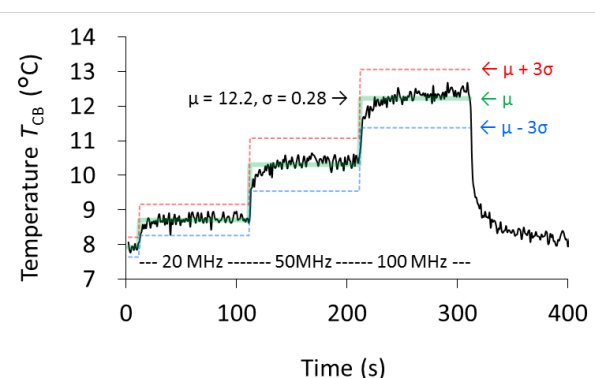


図 3-8 動作クロック周波数変化のチップ温度 (T_{CB}) への影響

それぞれの平均 μ から各要素が 3σ (99.7%) に含まれていることが確認できる

フィボナッチ数列計算

フィボナッチ数列はイタリアの数学者レオナルド・フィボナッチ (Leonardo Fibonacci) が 1202 年に考案した数列である。植物の花や実に現れる螺旋の数など、自然界の現象に数多く出現する。更に、コンピュータプログラミングでは再帰的処理のよい教材であるとされている。アセンブリ言語でのプログラミングも比較的容易であるため、発熱の基本的な挙動確認に選択した。n 番目の数列の値 F_n は式(3-1) で表され、表 3-7 に示す数列になる。

$$\begin{aligned} F_0 &= 0 \\ F_1 &= 1 \\ F_{n+2} &= F_n + F_{n+1} \quad (n \geq 0) \end{aligned} \quad (3-1)$$

表 3-7 Fibonacci 数列

bit 幅制限なし (左), 32bit 範囲 (中), 16bit 範囲 (右)

n	F_n	$F_n \pmod{2^{32}}$	$F_n \pmod{2^{16}}$
0	0	0	0
1	1	1	1
2	1	1	1
3	2	2	2
4	3	3	3
5	5	5	5
6	8	8	8
7	13	13	13
8	21	21	21
9	34	34	34
10	55	55	55
11	89	89	89
12	144	144	144
13	233	233	233
14	377	377	377
15	610	610	610
16	987	987	987
17	1,597	1,597	1,597
18	2,584	2,584	2,584
19	4,181	4,181	4,181
20	6,765	6,765	6,765
21	10,946	10,946	10,946
22	17,711	17,711	17,711
23	28,657	28,657	28,657
24	46,368	46,368	46,368
25	75,025	75,025	9,489
26	121,393	121,393	55,857
27	196,418	196,418	65,346
28	317,811	317,811	55,667
29	514,229	514,229	55,477
30	832,040	832,040	45,608
31	1,346,269	1,346,269	35,549
32	2,178,309	2,178,309	15,621
33	3,524,578	3,524,578	51,170
34	5,702,887	5,702,887	1,255
35	9,227,465	9,227,465	52,425
36	14,930,352	14,930,352	53,680
37	24,157,817	24,157,817	40,569
38	39,088,169	39,088,169	28,713
39	63,245,986	63,245,986	3,746
40	102,334,155	102,334,155	32,459
41	165,580,141	165,580,141	36,205
42	267,914,296	267,914,296	3,128
43	433,494,437	433,494,437	39,333
44	701,408,733	701,408,733	42,461
45	1,134,903,170	1,134,903,170	16,258
46	1,836,311,903	1,836,311,903	58,719
47	2,971,215,073	2,971,215,073	9,441
48	4,807,526,976	512,559,680	2,624

3.5.3 発熱プログラムの概要

次の段階では、PUにおける命令処理向けブロック内のデータパス (data path) の論理 0/1 レベル変化頻度に注目して、5つのグループの発熱プログラム (以下 HGP) を用意した。HGP は、高い発熱を得る単一命令あるいは複数命令列により構成されるプログラム、及び命令が処理するデータの明確化を目的とする。ただし、発熱の測定を容易にするため次の3つの制約を設定した。(1) 単一命令の実行比率は99%を超えること、(2) 発熱が分オーダで持続して安定すること、(3) 命令処理向けブロックの活性化と発熱の関係が説明できること。

HGP のプログラミング

FPGA デバイスは、1 Hz から 100 MHz までの動作クロックにおける HGP 実行により加熱された。PU 上では次の5群の HGP が実行され、加熱による温度変化が測定された。処理毎に 20 MHz, 50 MHz, 100 MHz の動作クロックで実行し、それぞれの周波数で 100 秒動作した平均値を図 3-9 に示す。

- ①動作後のビット変化のない単独命令
- ②ビット変化の少ない単独命令
- ③多数のビット変化を伴う単独命令
- ④複数の異なるデータに対して動作する小規模プログラム
- ⑤ボード上で FPGA 以外に積極的に電力を消費させる上記処理④

各 HGP で選択した具体的な対象命令またはプログラムは以下の通りである。① C.NOP 命令、②インクリメント命令、③ゼロから-1 と+1 を繰り返す命令、④32bit フィボナッチ数列計算プログラム、⑤プロセッサ内部のリソースデータを FPGA ボード上の7セグメント LED など表示デバイスに計算結果を表示するプログラム④である。④の候補として、バブルソート (Bubble sort) 及びユークリッドの互除法 (Euclidean algorithm) のプログラムも評価したところ、実測結果はフィボナッチ数列計算の温度変化が最大であった。

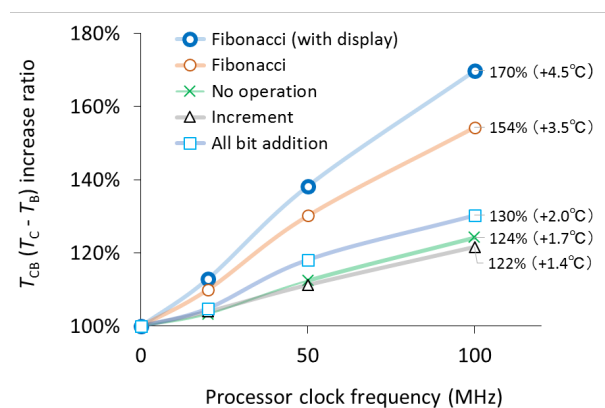


図 3-9 補正したチップ温度と動作周波数関係の HGP による差

処理①②③における単独命令（以後、ターゲット命令と呼ぶ）の実行では、高精度な発熱を実現する必要があった。また、各 HGP においてターゲット命令以外の命令の実行頻度を最小限に抑える必要があった。これらの必要性に対応するため、16bit 長の命令を 2,048 ワード格納した IM をターゲット命令で埋め、先頭の IM から順次実行させた。2,048 語の最後尾に IM の先頭へ分岐させる C.J 命令を配置する実行ループを用意した。この配置により、実行ループ内ではターゲット命令の実行頻度は 99.9%を超えた。IM の回路記述では HDL の case 文を使用し、default 条件を使う 1 行が IM をターゲット命令で埋めるようにした。

処理④,⑤におけるフィボナッチ数列の繰返し計算は以下のリストに示す Fibonacci_number_infinite_loop プログラムを用意した。リストはアセンブリ言語で記述し、リストにおける loop ラベルを含む実行ループ内（行 5~9 の間）では RF 内のレジスタ間データ移動、EX による演算、DM のリードアクセス、及びライトアクセス、PC の分岐制御を均等に実行するように命令を配置した（リストの“;”以降のコメント部を参照）。

```

1 | Fibonacci_number_infinite_loop:
2 |     C.LI  x8, 0x0
3 |     C.LI  x9, 0x1
4 |     C.LI  x10,0x2
5 | loop:  C.ADD x8,x9           ; EX
6 |     C.SW x10(0x3),x8       ; DM(write)
7 |     C.MV  x8,x9           ; RF
8 |     C.LW  x9,x10(0x3)      ; DM(read)
9 |     C.J   loop            ; PC

```

HGP の実行

ここでは、5 つの HGP の中で最も複雑なフィボナッチ数列の繰返し計算の実行結果を説明する。Fibonacci_number_infinite_loop プログラム実行中に機械語モニタが 1 クロック毎のトレースによりプロセッサの内部資源データを端末ソフトウェアに表示した文字コードを表 3-8 に示す。左より (1) クロック数 (clk), (2) PC の値 (address), (3) 命令コード (16 進数と 2 進数の OP code), (4) 命令の種類 (instruction), 及び (5) 機械語のニモニック (mnemonic) が表示されている。上述のリストの 2 行目が clock 1 からはじまり (①行), 9 行目が clock 8 の C.J 命令の実行により 5 行目の clock 9 で loop ラベルに分岐してループを構成したこと (②行) を示している。以後, ③行, ④行

と C.J 命令がループを構成している. 単一サイクルのマイクロアーキテクチャの採用により分岐命令 C.J も 1 クロックで実行が完了していることが分かる. なお, アセンブリ言語によるプログラムは 1 行アセンブラ (one-line assembler) を用意し, 1 命令ずつ機械語に変換した (図 3-10).

表 3-8 Fibonacci 数列計算プログラムのトレース結果

	clock	address	OP code	instruction
①	0	00000424	: 9426 = 10010100_00100110	(9) C.ADD
	1	* 00000620	: 4401 = 01000100_00000001	(8) C.LI
	2	00000622	: 4485 = 01000100_10000101	(8) C.LI
	3	00000624	: 4509 = 01000101_01001001	(8) C.LI
	4	00000626	: 9426 = 10010100_00100110	(9) C.ADD
②	5	00000628	: C540 = 11000101_01000000	(B) C.SW
	6	0000062A	: 8426 = 10000100_00100110	(8) C.MV
	7	0000062C	: 4544 = 01000101_01000100	(A) C.LW
	8	0000062E	: BFE5 = 10111111_11100101	(E) C.J
	9	00000626	: 9426 = 10010100_00100110	(9) C.ADD
③	10	00000628	: C540 = 11000101_01000000	(B) C.SW
	11	0000062A	: 8426 = 10000100_00100110	(8) C.MV
	12	0000062C	: 4544 = 01000101_01000100	(A) C.LW
	13	0000062E	: BFE5 = 10111111_11100101	(E) C.J
	14	00000626	: 9426 = 10010100_00100110	(9) C.ADD
④	15	00000628	: C540 = 11000101_01000000	(B) C.SW
	16	0000062A	: 8426 = 10000100_00100110	(8) C.MV
	17	0000062C	: 4544 = 01000101_01000100	(A) C.LW
	18	0000062E	: BFE5 = 10111111_11100101	(E) C.J
	19	00000626	: 9426 = 10010100_00100110	(9) C.ADD
⑤	20	00000628	: C540 = 11000101_01000000	(B) C.SW
	21	0000062A	: 8426 = 10000100_00100110	(8) C.MV
	22	0000062C	: 4544 = 01000101_01000100	(A) C.LW
	23	0000062E	: BFE5 = 10111111_11100101	(E) C.J

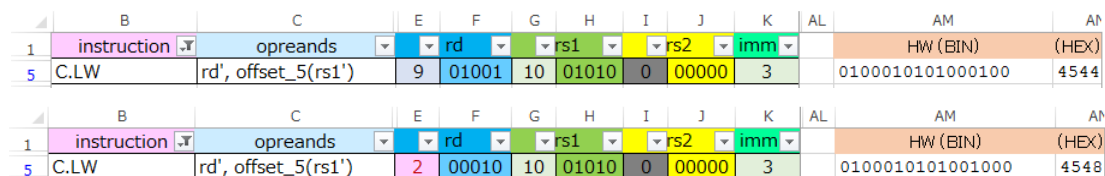


図 3-10 1 行アセンブラ操作の様子

instruction セルから命令を選択し, rd, rs1, rs2, imm セルにオペランドとなる数字を入力することにより HW (BIN) (HEX) に機械語が生成される (上), rd オペランドがエラーであることを示す (下)

参照 HGP の選択

図 3-11 は、クロック周波数 20 MHz, 50 MHz と 100 MHz で 100 秒間実行した 5 つの HGP のそれぞれについての T_{CB} の平均値を示したものである。5 つの HGP すべてにおいて、クロック周波数に対する T_{CB} の変化は原点を通る線形近似であり、決定係数 R^2 は 0.98 を超えていた (表 3-9)。したがって、変化率は、変化を線形関数とみなして、図 3-11 にパーセンテージで表現した。変化率が 170%と最も大きいフィボナッチ数列計算プログラムを、以後の消費電力に関する実験にて参照 HGP として選択した。

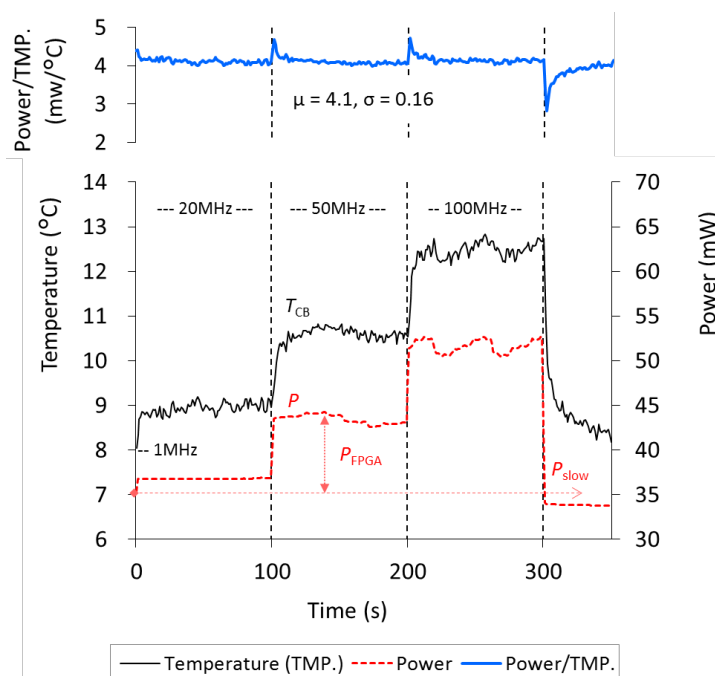


図 3-11 Fibonacci 数列計算時の温度と消費電力の変化
温度と消費電力 (下) とそれらの比 (上)

表3-9 チップ温度 (T_{CB}) と動作周波数関係近似のHGPによる差

HGP	0 - 100MHz 温度上昇率 (%)	一次式近似時の 決定係数 R^2
No operation	124	0.999
Increment	122	0.995
All bit addition	130	0.984
Fibonacci	154	0.995
Fibonacci (with display)	170	0.996

3.6 発熱と消費電力との関係モデル

ここでは、HGP 実行による発熱を反映する T_{CB} と、FPGA デバイスが消費した電力の関係について説明する。

3.6.1 FPGA デバイスの消費電力算出方式

図 3-3 に示した FPGA ボードの消費電力 P は 5.5V に固定した供給電源電圧 V_{in} とシャント抵抗 (R_{shunt}) の両端電圧 V_{drop} から式(3-2)で FPGA ボード全体の概算値として表現される。シャント抵抗には許容差 1% のリード線形抵抗器を使用し、表示 1.00Ω の R_{shunt} 値を 4 線測定方式で実験直近の 24 時間測定した平均から 0.99Ω とした。

$$P = (V_{in} - V_{drop}) \frac{V_{drop}}{R_{shunt}} \quad (3-2)$$

FPGA のみの消費電力を表す消費電力 P_{FPGA} は 式(3-3)で表され、十分に遅い 1 MHz でのプロセッサ実行時の消費電力 P_{slow} を差し引いて P からの補正により P_{FPGA} が得られる。

$$P_{FPGA} = P - P_{slow} \quad (3-3)$$

図 3-11 の下段は、補正前の消費電力 P と T_{CB} を示している。図 3-11 の上段は、補正後の消費電力 P_{FPGA} と T_{CB} の比率を示している。この比率は、動作クロックの周波数切替え直後を除いて良好に安定しており、 T_{CB} と消費電力の変化率は $4.1 \text{ mW}/^\circ\text{C}$ であった。この結果から、 T_{CB} から線形近似による消費電力推定の有効性が示された。ボード温度による補正したチップ温度 T_{CB} は、提案したプロセッサハードウェアとプログラムとデータで構成されるソフトウェアの組み合わせによって発生するチップ内の発熱量を検知したものである。

3.6.2 消費電力と関連温度検討の課題

まず、3.5.7 で述べたように、実験に選択したクロック周波数は 20 MHz, 50 MHz, 100 MHz のみであり、消費電力と関連温度の連続的な関係のさらなる明確化には、測定周波数を追加した実験が必要である。

次に、3.5.5 で述べたように、実験した 5 組の Hot program と Hot data に、より高い発

熱を期待できる処理プログラムと演算データを追加していく必要がある。

更に、加算器やメモリを含むプロセッサ内の機能レベルで、出力状態を変化させる入力データ列の理解も追加対象の一つである。図 3-12 に 32bit 加算器に 1,000 回連続したデータ列を入力して得られる加算結果と、1 回前の加算結果の 2 データ間で変化する bit 数の推定量を示す。bit 数には和 (sum) だけでなく桁上げ (carry) が含まれ、推定の精度を高めた。入力データ列には (a) インクリメント数 (1 ずつ増加する数列), (b) マイクロプロセッサの命令コード (RISC-V ベース ISA をもつプロセッサの実プログラムから抽出), (c) 擬似乱数 (Linear Feedback Shift Register : LFSR による), (d) Fibonacci 数列を用い、推定の計算には Visual Basic 言語を用いた。図 3-12 には、4 種の指標を示す。(1) Σ は変化 bit の総量, (2) μ は変化量の平均, (3) 標準偏差 σ は変化量の揺らぎ幅, (4) $\sigma\Delta$ は隣接する 2 変化量差の揺らぎ幅を示す。これら 4 指標すべてが大きいデータ列で加算するほど、継続的に高い発熱が示唆される。ただし、これらの指標の重み付けと生成の難易度については今後も議論を重ねる必要があるものの、4 指標すべてで Fibonacci 数列の優位性が示されている。

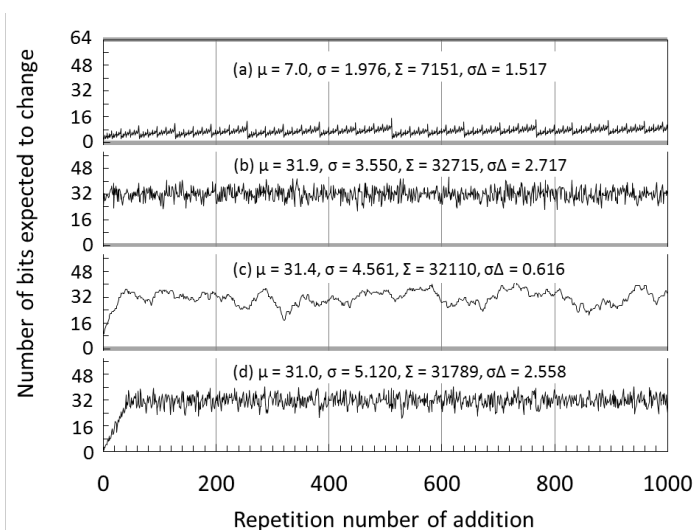


図 3-12 連続した加算で変化する bit 数

最後に、今後取り組むべき本研究の課題について述べる。課題には、外部環境の反映がある。(1) ボード上の他熱源の存在, (2) システム全体のコールドスタート, (3) ターゲット FPGA の自然放熱, (4) FPGA を含むボード周辺の強制放熱である。(1) に関連して、実験に使用した FPGA ボード表面からと裏面からの可視画像と IR 画像を、それ

ぞれ図 3-13 と図 3-14 に示す. 本章で使用した FPGA ボードで評価対象の FPGA (IR 画像の中央付近) と同規模で発熱する IC (IR 画像の縦横マトリクス B3 付近) が動作している状態を示す. エポキシ樹脂や配線銅箔を含むボード材料を伝導し FPGA のチップ上で平衡した熱を CTS は温度として感知していると考えられる. このように, ボードに実装された状態での熱管理が複雑で, 精度を高めるには複数の解決すべき課題が残っている.

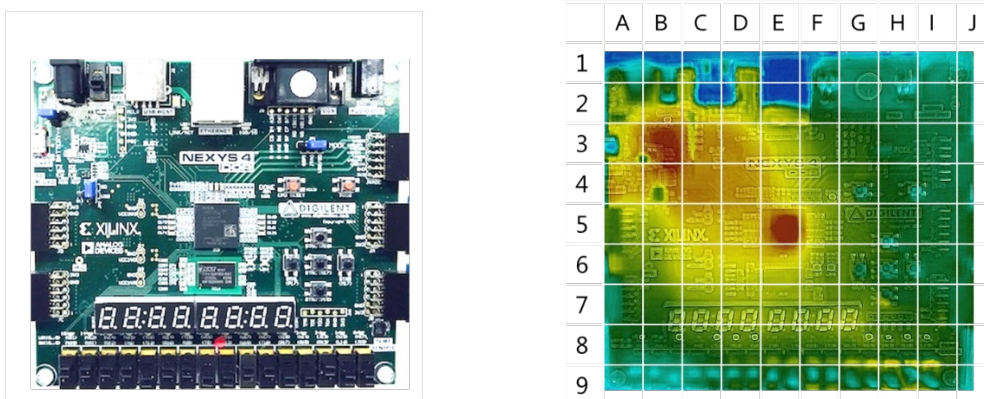


図 3-13 FPGA ボード写真 (表面)
(左) 可視画像, (右) IR 画像

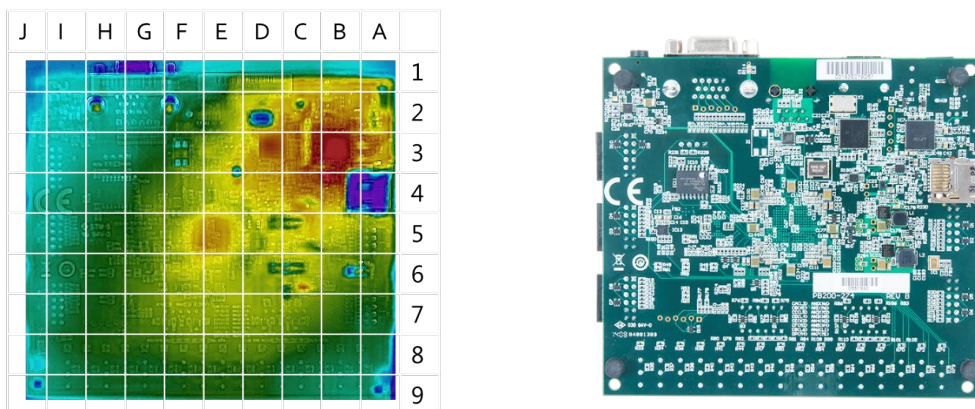


図 3-14 FPGA ボード写真 (裏面)
(左) IR 画像, (右) 可視画像

3.7 おわりに

以上で述べたように、本章では FPGA 上に実装した小規模プロセッサについて、ボード温度とボード外の周囲温度でチップ温度を補正する電力推定関係を導いた。今後はこれらの測定領域の拡大とデータを追加する。実験により導いた推定関係の精度を向上させるためには、測定範囲の拡大やデータの追加が有効である。

第4章 FPGA チップ加熱用リングオシレータの実装と評価

4.1 はじめに

2022年下半期には3nmプロセスのICチップ量産が始まると伝えられている。10nm以下プロセスを採用したマイクロプロセッサ (Microprocessor : MPU) とシステムオンチップ (System on-a Chip : SoC) の量産がはじまった。20nm以下プロセスを採用したFPGAデバイスもIC市場に広がりつつある中、IC応用製品の熱設計・熱管理がますます重要になっている [40]。内部配線材料やパッケージ材料、プロセス技術などには150°Cのジャンクション温度を限界とする材料が選択され、更に構造や製造工程などが設計されてきた。限界を超えぬよう、製品の動作環境と要求信頼性に応じたジャンクション温度の管理が不可欠である。更に、設計時だけでなく運用時の熱管理が必須となる。熱管理には、基準温度への加熱能力と到達温度の測定能力が必要である。温度測定用に専用の温度センサを内蔵するFPGAが増えつつある [31]。一方、専用の加熱器はFPGAには内蔵されておらず、加熱は運用向け回路の消費電力に依存している。

本章はリングオシレータ (以下RO) 出力の発振周波数とともに配線路長を制御する拡張駆動加熱器 (以下ROH) を提案する [50]。ROHはFPGAに実装される運用向け回路を代替し、RO出力の配線路長拡張に向ける。その結果増大した発熱をもってして、チップ加熱を高温BISTに資するROである。ROHを低消費電力FPGAに実装し、既報の高速・高性能FPGAと同程度の加熱特性を得た。

以下、本章は次のように構成される。まず、4.2でROの関連研究について議論する。4.3ではFPGAチップ発熱の概念を説明して、提案するROHを紹介する。つぎに、4.4でROH評価のための実験用回路と加熱評価システムのXilinx社Artix-7ファミリへの実装について説明する。4.5ではチップ加熱実験で得られた結果を示す。更に、4.6でROHの応用例を述べる。最後に、4.7で本章を締めくくる。

4.2 リングオシレータと関連研究

RO は発振器の一つであって、反転遅延素子をリング状に負のフィードバックをかけた回路を構成し、全素子の総伝播遅延時間を利用する。代表的な反転遅延素子はインバータである [51]。図 4-1 に 3 種類の RO の回路を示す。(a) 最も単純なインバータ 1 段の RO, (b) 論理的にはインバータ 1 段と等価で遅延素子に 1 段のバッファを加えた RO, (c) 発振出力 Q_5 を持つ RO 回路では、遅延素子とするインバータ INV と 4 段のバッファ B1, B2, B3, B4 をリングに接続する RO である。

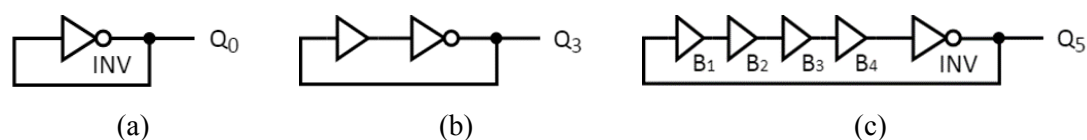


図 4-1 リングオシレータ (RO) の回路例

RO は CR 素子やセラミック振動子、水晶発振子など半導体デバイス以外の部品コストを要しないものの、周波数の発振精度や長時間の安定性の低さと、温度や動作電圧に対するジッタなどの課題を伴う。これらの課題解決に向けては、FPGA の回路と配置・配線の規則性から RO によるチップ温度推定の精度を高める研究がある [52], [53]。逆にこれらの課題を利用して、RO のジッタを使いセキュリティ回路を攻撃する実装研究なども注目されている [54]。

このように、RO を発振器として発振信号の特性を追及する研究とともに、FPGA チップの加熱器として RO が重要になってきた [55], [56]。[55] では、基本加熱器 Basic Heater : BH として複数種の非同期加熱器と同期加熱器を比較評価した。加熱能力は 1 段遅延の RO が高く、加熱に対する電力効率も同期インバータ列が高かった。[56] では、1 段遅延で小規模 (たとえば、10 組以下の) RO による基本加熱器 Self-Heating Element : SHE のチップ内配置方式が複数提案された。SHE が設定したチップ温度へ加熱能力を実証した。[55], [56] はともに高速・高性能な FPGA (65nm プロセスを採用した Xilinx 社の Virtex-5) に 1,000 組オーダの基本加熱器を実装した。BH, SHE とともに 100 秒オーダの時間をかけて、チップを 125°C 以上に加熱した。高温環境における BIST 用恒温槽の排除が期待できる結果である [57], [58]。

一方、恒温槽は多数の被試験チップ (Device Under Test : DUT) を 100 リットル超えの槽内空気で 100°C 以上に加熱する [59]. このため、設置に要する容積、及び稼働に必要な電力を確保しなければならない. とりわけ、一つだけのデバイスや運用中のデバイスの加熱にとって、設置場所・稼働環境での制約が高い障壁となる. 加えて、槽内の温度変化は分/°C オーダの時間であり、短時間での BIST 実行を妨げる [60].

また、[55], [56] で実証された 65nm プロセス FPGA は、市場では 45nm → 28nm プロセス FPGA へ置換えが進み、さらに 20nm 以下プロセス FPGA の市場投入がはじまっている. FPGA では依然として微細化が進んでおり、トランジスタばかりでなく配線での信号伝播遅延・消費電力への影響比率が高くなっている [61]. 「スケーリング則」 (scaling law または Dennard scaling) として知られる消費電力低減も考慮し、加熱能力と電力効率について RO の基本加熱器の構造・実装の見直しが不可欠である. [56] では、加熱について有益な実験結果が示されているが、温度を維持する制御については明らかにされていない.

本章では、最近の市場で主流な低消費電力 FPGA を対象にして、拡張駆動 RO 加熱器の回路・実装方式を提案する. そして、運用環境における高温 BIST で必要なチップ加熱能力に資することに貢献する.

4.3 拡張駆動リングオシレータ加熱器の提案

4.3.1 FPGA の進化と懸案事項

近年, Programmable Logic Device : PLD の一種である FPGA の応用範囲が拡大している. 人工知能 (AI) を支える深層学習手法の高性能処理が, 代表的応用といえる. また, 規則性の高いアーキテクチャと回路方式, レイアウト構造を持つ FPGA は, プロセス技術進歩の恩恵を受けやすい IC である. これまでほぼすべてのプロセス技術毎の FPGA が市場に投入されてきている.

一方, 懸案される課題には設計者・利用者の責任範囲の拡大がある. IC の動作限界を定める絶対最大定格の一つである最大ジャンクション温度 ($T_{j(\text{MAX})}$) は, 限定的な例外を除き, およそ半世紀前の商用トランジスタと同等である [62], [63], [64]. メーカーの違いやプロセス技術の変遷にかかわらず, FPGA においてもこの状況は同様といえる. $T_{j(\text{MAX})}$ を超えると, 半導体デバイスとしての動作不良だけでなく, 不可逆的なパッケージ変形や破壊による一時的あるいは永続的な不良が発生する. この不良は FPGA 応用製品における異常動作を招く原因となる. PLD である FPGA の出荷時には, 回路プログラムだけでなくユースケースも確定していない. このため, メーカーは周囲温度またはパッケージ温度による動作保証ができず, 利用者が責任の一部を負担しなければならない. この懸案への対策に備えて, 温度センサと温度監視, 制御機能で構成される保護回路を内蔵する FPGA が製品化されている.

また, メーカーは内製部品と開発環境を用意するが, 設計者の論理設計・論理検証によりデバイス品質が作り込まれる. 設計保証 (design assurance) として知られている品質保証手続きに適用できる内製部品には, 知的財産権をメーカーが保有する IP コアが含まれる. したがって, メーカー以外の第三者が供給する部品を実装する場合, 設計者がデバイス品質の保証責任を一部負担する必要がある.

4.3.2 IC の熱発生

IC チップが発熱する原因はチップ内で電力を失うこと、すなわち消費電力の発生による。FPGA に限らず CMOS IC の消費電力 P は静的電力成分 P_{static} と動的電力成分 P_{dynamic} 、漏れ電力成分 P_{leak} を合計して式(4-1)で示される。

$$P = P_{\text{static}} + P_{\text{dynamic}} + P_{\text{leak}} \quad (4-1)$$

漏れ電力と静的電力の合計（待受け電力）は 100nm プロセス FPGA で 100 mW を超える程度であったが、28nm プロセス FPGA で 1,000 mW 近くになる。一方、動的電力は個別 RO の発振出力周波数に応じて、個別動的電力の総和で示される。FPGA を構成する論理ブロックと配線ブロックを発振出力が伝播する際に、主たる個別動的電力成分が発生する。論理ブロックは複数の LUT が基盤構造であり CLB や LE (Logic Element) などと呼ばれる。配線ブロックは配線チャネルと配線スイッチを組み合わせで構成される。近年のプロセス技術による配線ブロックでは、信号が伝播する配線路における伝播遅延とともに、動的電力の影響が増している [65]。本章では、配線ブロックで発生する動的電力の発熱への影響を重視する点が、以前の研究と異なる着眼点である。このため、段階的には次の条件を明確にする。(1) 1 ないし少ない論理ブロックを使い発熱を最大化する発振、(2) 複数の論理ブロックを使い発熱を最大化する発振出力の伝播、(3) 利用可能なブロック数内で発熱を最大化する (1), (2) の組合せ。

4.3.3 リングオシレータ加熱器

ここでは、利用者が FPGA 応用製品の運用を開始した後に、設計者が実装した回路プログラムの温度特性試験に向けて、FPGA 内部を加熱する ROH を提案する。目標温度までチップを加熱する回路プログラムを実装する ROH は、運用向け回路プログラムの一部あるいは全部を柔軟に代替する。ROH を実用に供するにはチップ加熱に対する (1) 発振出力の周波数、(2) 発振出力の配線路長、(3) 規模の影響を明確にしておく必要がある。このため、加熱により到達するチップ温度を測定変数とし、以下 (1), (2), (3) に示す制御変数との関係を測定する。

- (1) 遅延素子の遅延段数
- (2) 遅延素子の駆動段数
- (3) モジュール数

4.4 FPGA への実装

4.4.1 ゲーテッド遅延素子

ここでは、ROH の基本構成要素となるゲーテッド遅延素子（以下 GD）を説明する。GD は RO 内遅延素子の信号伝播を AND ゲートや OR ゲートで制御する。図 4-2 は 2 入力マルチプレクサを構成する GD である。遅延対象の入力 in と fb は、選択信号 $gate$ に応じてゲート G_1, G_2, G_3 からそれぞれの遅延時間 $t_{pG1}, t_{pG2}, t_{pG3}$ を伴って出力 out に伝播される。つぎに、図 4-3 (a) に N 段の GD で構成する RO を示す。 n 段目となる GD_n からの出力信号 out は、 $n-1$ 段目となる GD_{n-1} の入力信号 in に接続される。リング先頭である GD_0 の出力が接続されるインバータ出力が発振出力信号 $out \#xxx$ となる。また、発振動作の許可信号 $enable$ で $out \#000$ をマスクした fb 信号は、リング最後尾である GD_{N-1} の入力にも接続される。 fb 信号を選択した GD_n から前方段である $GD_n \sim GD_0$ は、発振の役割を持つ遅延段として動作する。一方、 GD_{n+1} から後方段である $GD_{N-1} \sim GD_{n+1}$ は、動的電力発生の役割を持つ駆動段として動作する。図 4-3 (b) に GD 段数を 4 に選択した RO の等価的な構成を示す。この構成は図 4-1 (c) に示した RO 回路と同じ 5 遅延素子を持つ。加えて、拡張駆動段として $N-4$ 段の遅延素子を持つ。

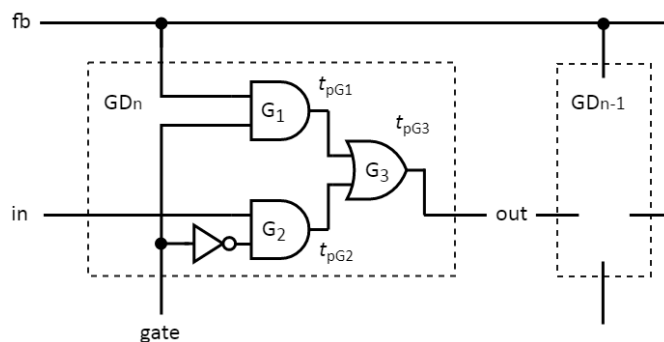


図 4-2 ゲーテッド遅延素子 (GD) の回路

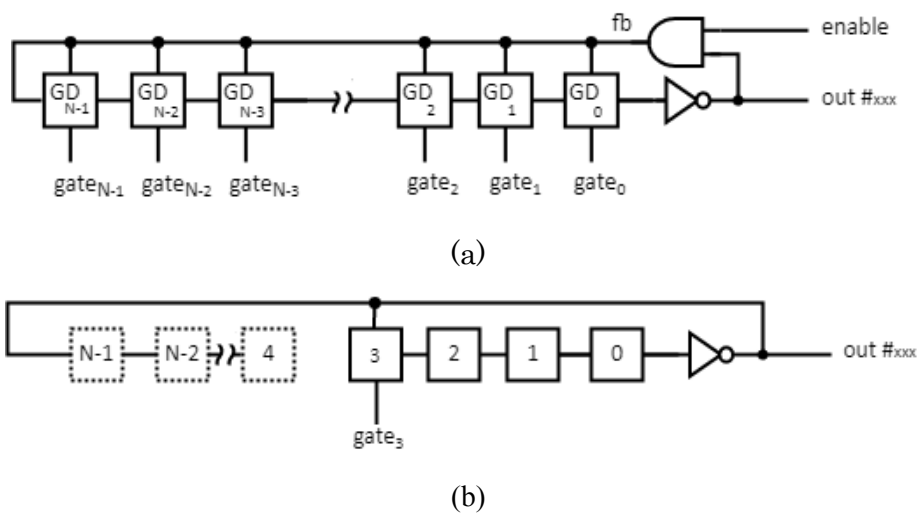


図 4-3 ゲーテッド遅延素子 (GD) で構成した提案 RO の構成

4.4.3 加熱評価システム

図 4-5 に ROH を実装した加熱評価システムの構造を示す。なお、GD 段数 N は回路合成時に指定した。つぎに、ROH を除くブロックの機能と動作を説明する [66]。PU は温度特性のターゲットとする組込用途向け 32bit MPU である。PU の ISA はオープンな RISC-V であり、標準 ISA に命令セット拡張の一部を加えて構成した [47]。ここでは、RISC-V で最小となる 16bit 命令語長の圧縮型命令セット拡張 ("C" Standard Extension : RV32C) のうち 27 命令を選択した。マイクロアーキテクチャにはパイプライン段数を 1 とする単一サイクル方式を採用している。

iMCM は PU 用機械語モニタ機能を提供する. 共有メモリ (CM) は PU と iMCM 双方からアクセスされる. IO は LED 表示, UART 通信など FPGA 以外のデバイスを制御する. クロックシーケンサ (CS) は ROH の 3 制御変数と PU の動作クロック (最高 100 MHz) を, あらかじめ最大 1,024 ステップ×256 エントリのパターンを登録した順序で他ブロックへ供給する. 表 4-1 に CS に実装した具体的な機能を示す. プラットホームモニタ (PM) はリアルタイムクロック IC (RTC) が刻む 1 秒毎に全温度センサのサンプリング情報をシステム外部へ通知する. 熱制御器 (TC) は ROH と温度センサからのデータを基に, あらかじめ設定した目標温度にチップを加熱した温度を維持する. タイマ (TU) は評価システム内のタイミング信号を発生する.

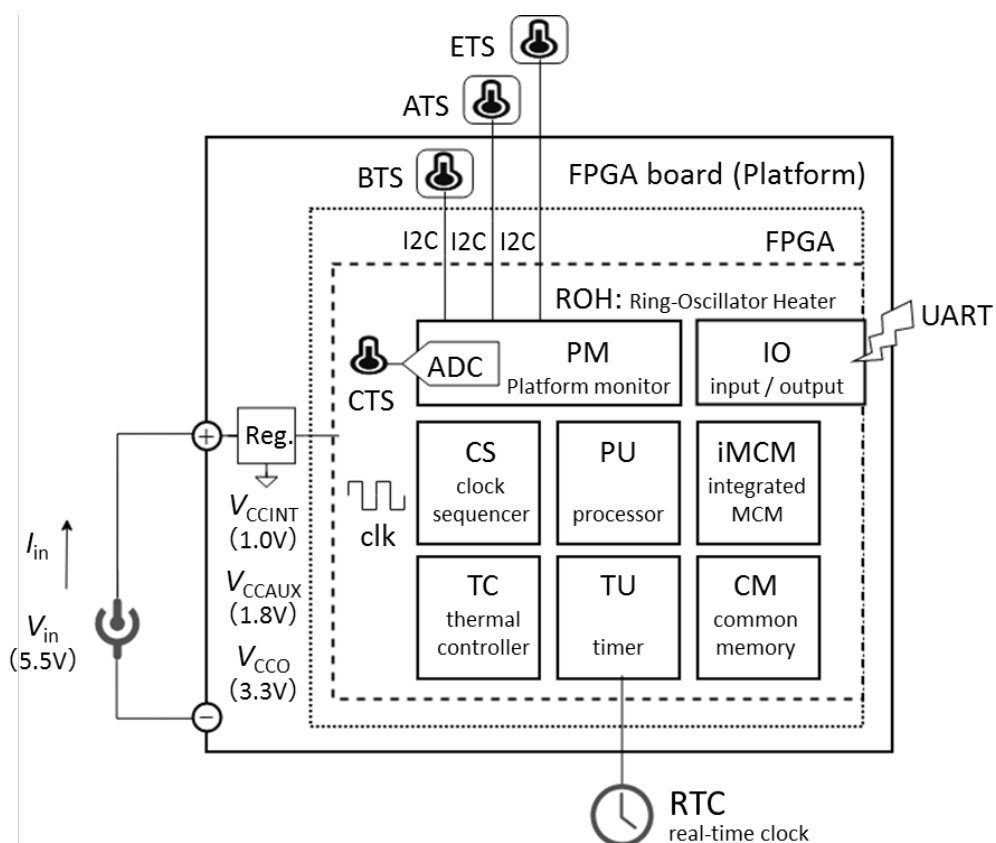


図 4-5 加熱評価システムの構造

表4-1 クロックシーケンサ (CS) の主な機能

項目	仕様
エントリ	256
エントリ構成	1,024 ステップ / エントリ (max.)
ステップ指定パラメータ †	
持続時間 (s)	10 / 20 / 50 / 100 / 200 / 500 ms, 1 / 2 / 5 / 10 / 20 / 50 / 100 / 500 100 s
クロック周波数 (Hz)	0 / 1 / 10 / 100 Hz, 1 / 10 / 100 KHz, 1 / 10 / 11.1 / 12.5 / 14.3 / 16.7 / 20 / 25 / 33.3 / 50 / 100 MHz
ROH関連パラメータ ††	活性化 : on / off モジュール数 : 0~10 遅延段数 : 1, 2, 3, 4, 8

† 12bitで構成, †† 4bitに圧縮している

4.4.4 温度センサ

ここでは、システムに搭載した温度センサについて説明する。温度センサはオンチップ温度センサとオフチップ温度センサに分類される。チップ温度を測定するオンチップ温度センサ (CTS) は FPGA 内でチップ上に組み込まれる。アナログ値として感知されたチップ温度データはアナログ - デジタル変換回路 (ADC) で量子化され、デジタル値が FPGA 内に取り込まれる [67]。オフチップ温度センサの目的は、外部熱源からの熱伝導と外部放熱先への熱対流がチップ温度に与える影響を観測することである。3組のオフチップ温度センサは (1) FPGA デバイスから最も離れた位置でボード温度を測定するボード温度センサ (BTS), (2) ボードから数 cm ほどの距離で周囲温度を測定する周囲温度センサ (ATS), (3) ボードから 10cm ほどの距離でボードの発熱に依存しない周囲温度を測定する拡張温度センサ (ETS) である。BTS, ATS, ETS にはセンサと一体化した ADC を集積する Analog Devices 社の温度センサ IC (ADT7410 と ADT7420) を使用した。それぞれの IC が測定したデジタル値は、I-squared-C : I2C バスを介して FPGA に入力される。CTS, BTS, ATS, ETS が感知して変換したデジタル値の温度を T_c , T_b , T_a , T_e で表す。なお、CTS は $0.123^{\circ}\text{C}/\text{LSB}$, $\pm 4^{\circ}\text{C}$, BTS, ATS, ETS は $0.078^{\circ}\text{C}/\text{LSB}$, $\pm 0.5^{\circ}\text{C}$ の分解能と精度を持つ。測定された温度は UART 通信を介してホスト PC に通知され、端末ソフトウェアが受信する。

4.4.5 FPGA の選定

現在、20nm 以下プロセスを採用した FPGA デバイスが IC 市場に広がりつつある中、これらの先端デバイスにはプロセス技術を高性能化・高機能化に活かすものが非常に多い。本章では 28nm プロセスを採用した Xilinx 社の FPGA である 7 シリーズのうち、消費電力が最も低い Artix-7 ファミリを選定した [46]。Artix-7 ファミリは規模の異なる 8 デバイスで構成されている。本章では中規模デバイスである XC7A100T を使用した [45]。

また、実験の容易性と再現性を考慮して FPGA プラットホームには Digilent 社の NEXYS4-DDR ボードを放熱・空冷なしの構成で使用した。このボードに搭載されるデバイス品名 XC7A100T-1CSG324 の $T_{J(MAX)}$ は 125°C で、動作保証されるジャンクション温度（推奨動作温度）は 85°C 以下と定められている。このため、 T_c が 125°C を超えぬよう FPGA の保護回路を有効にして実験した。

すべての回路は Verilog HDL 記述を用いて回路合成した。回路合成と配置・配線を含む FPGA 開発環境には Xilinx 社の Vivado ツール (v2016.2) を使用した。Vivado は FPGA 設計フロー全体をカバーするソフトウェア・ツールの集合である。HDL 記述から FPGA デバイスに回路プログラムするビットストリーム (bitstream) の生成までを次の 4 つの段階で処理する (図 4-6)。(1) Verilog HDL または VHDL 記述からの回路合成、(2) 配置 (実装の前半)、(3) 配線 (実装の後半)、(4) ビットストリームの書込み (.bit データの生成)。それぞれの段階で回路部品 (Intellectual Property : IP) と、実装上の制約 (constrain)、設計ルール (Design Rule Check : DRC)、回路プログラムする FPGA デバイスの端子情報などを参照する。各段階の処理状態は (1) エラー (error)、(2) 致命的なワーニング (critical warning)、(3) ワーニング (warning) にレベル分けされた上で検出・記録される。FPGA に正しくない回路がプログラムされる不具合を防ぐため、エラーが記録されると次の段階には進まない。また、FPGA には 100 MHz のクロック信号が供給される。ただし、本章ではタイミング解析に不可欠なクロック制約と配置・配線に関する制約を設定せず、設計保証がない条件で回路実装した。

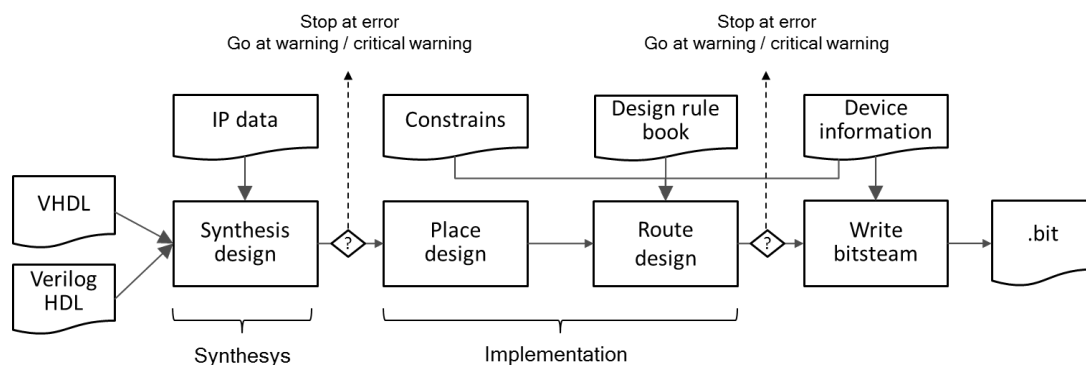


図 4-6 Vivado 開発環境の基本処理フロー

Xilinx 社に限らず、また開発環境用に留まらず、商用のソフトウェアはバグの修正と機能・性能の改良・改善のために頻繁に更新された版が公開される。Vivado ツールは v2016.2 以降 v2020.1 までおよそ 3 回/年、計 11 回改版されてきた。本研究では版数を固定 (version fix) し、回路合成と配置・配線の設計方策 (design strategy) 変更による FPGA リソース占有率の差異発生を避けた。また、版数の固定はエラー類の検出と取扱いの差による設計フロー指定の変更も避けた。RO の回路記述は、開発環境にとっては潜在的なエラーとして次の処理がなされる。(1) 偶数個のインバータの直列接続が回路の最適化によりインバータを 2 つずつ削減する、(2) 組合せ回路出力の入力への直接接続の存在を combinational loop エラーとして記録する。(1) は論理合成の段階で削減される。また、(2) は配置・配線の段階で記録され、引き続きビットストリーム生成段階で設計フローが中断される。これらの障害への対応として、以下の対策を講じた(図 4-7)。

(1) 論理合成の Verilog HDL 記述にて組合せ回路の出力を最適化対象から外す記述を加える。GD や ROH モジュールなどループを構成する回路の出力信号 loop_out_xxx を定義する wire 宣言に、keep 属性を

```
(* keep = "true"*) wire loop_out_xxx;
```

のように追加した。この追加により、loop_out_xxx 信号が他の組合せ回路に吸収されるような最適化を防いだ。

(2) combinational loop の記憶レベルをエラーからワーニングに置き換える。Vivado がサポートするインタープリタ型プログラム言語である Tool Control Language : TCL スクリプト

```
set_property SEVERITY {warning} [get_drc_checks LUTLP-1]
```

を用意した。ビットストリームの書き込み段階に進む直前にこのスクリプトを実行し、combinational loop エラーの発生を示す“LULTP-1”を含むエラー記録をワーニングに置き換えることにより、設計フローの中断を回避した。

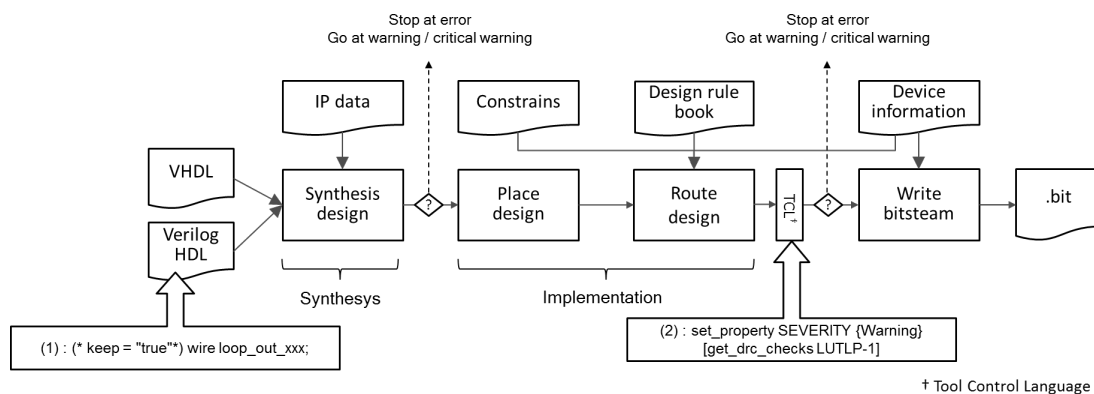


図 4-7 ROH 実装でカスタマイズした処理フロー

4.4.6 実装結果

加熱評価システムを構成するブロック単位の使用リソースとしての LUT 数と、実装した中規模 FPGA デバイス (XC7A100T) での占有率を表 4-2 に示す。また、GD 段数を 2^3 毎に 8~64 に変化させた場合の LUT 数と占有率を表 4-3 に示す。単一 GD 段あたりおよそ 1×10^3 の LUT を占有している。ROH の占有 LUT 数と ROH 以外の占有 LUT 数との合計が利用可能 LUT 数を超えぬよう、すなわち占有率が 100%に達しないように GD 段数を選択した。表 4-4 に小規模 FPGA デバイス XC7A35T での占有率を、そして表 4-5 には大規模デバイス XC7A200T の占有率を示す。初期の実装では XC7A35T で選択できる最大 GD 段数 16 で実装した。このように、小規模 FPGA デバイスでは加熱に必要な GD 段数選択が制限を受ける可能性がある。加えて、破壊に及ぶ不良が発生する可能性から、ROH モジュール活性時に T_j (MAX) に不用意に届かぬ GD 段数選択の考慮が必要である。

表4-2 加熱評価システムの各FPGAリソース占有率

Block	LUTs	Utilized Ratio (%)
PU: Processor Unit	5,979	9.4
CM: Common Memory	545	0.9
iMCM: integrated MCM	2,394	3.8
CS: Clock Sequencer	116	0.2
PM: Platform Monitor †	3,119	4.9
IO: Input / Output	486	0.8
TC: Thermal Controller	56	0.1
Sub total	12,694	20.0
ROH (# of GD stages=16)	17,829	28.1
Total: 63,400 (all)	30,523	48.1

† including TU.

表4-3 加熱評価システムの各FPGAリソース占有率
(中規模FPGAデバイスFPGA-m 実装におけるROH GD段数の感度)

Block		LUTs	Utilized Ratio (%)	Total (%)
Blocks except ROH		12,694	20.0	—
ROH	8	9,829	15.5	35.5
	16	17,829	28.1	48.1
	24	25,868	40.8	60.8
	32	33,836	53.4	73.4
	40	41,868	66.0	86.1
	48	49,868	78.7	98.7
	56	57,872	91.3	111.3 [†]
	64	65,868	103.9 [†]	123.9 [†]
Total: XC7A100T (FPGA-m)		63,400	100	

[†] exceeds size of total LUTs in FPGA-m.

表4-4 加熱評価システムの各FPGAリソース占有率
(小規模FPGAデバイスFPGA-s 実装におけるROH GD段数の感度)

Block		LUTs	Utilized Ratio (%)	Total (%)
Blocks except ROH		12,694	20.0	—
ROH	8	9,829	47.7	109.3 [†]
	16	17,829	86.5	148.2 [†]
	24	25,868	125.6 [†]	187.2 [†]
Total: XC7A35T (FPGA-s)		20,600	100	

[†] exceeds size of total LUTs in FPGA-s.

表4-5 加熱評価システムの各FPGAリソース占有率
(大規模FPGAデバイスFPGA-l 実装におけるROH GD段数の感度)

Block		LUTs	Utilized Ratio (%)	Total (%)
Blocks except ROH		12,694	20.0	—
ROH	8	9,829	7.3	16.8
	16	17,829	13.3	22.8
	24	25,868	19.3	28.8
	32	33,836	25.3	34.7
	40	41,868	31.2	40.7
	48	49,868	37.2	46.7
	56	57,872	43.2	52.7
	64	65,868	49.2	58.6
Total: XC7A200T (FPGA-l)		134,000	100	

4.5 チップ加熱実験

4.5.1 予備実験の結果

ここでは、加熱評価システムの回路プログラム結果が FPGA へダウンロードされ、CS が待機状態中に温度センサから通知された測定結果について説明する。CS は 4.4.3 で述べたように (1) PU の動作クロック、(2) ROH 関連パラメータ、及び (3) これらのパラメータの持続時間を 1 組とするシーケンスのエントリを最大 256 組記憶し、ボード外部からエントリの指定とシーケンスの開始を指定した。CS の実装により、数時間オーダの測定を半自動化した。表 4-6 に CS に組み込んだ主なシーケンスを示す。

表4-6 クロックシーケンサ (CS) に組み込んだ主なシーケンス

名称	目的	活性化 モジュール	活性化時間
CS000	ROH発振周波数モニタ	10	10秒×1,000 [†]
CS001	CSの初期動作確認	0	10秒×4 ^{††}
CS002	基本発熱確認 (短時間)	1~10	10秒×10
CS003	基本発熱確認 (長時間)	1~10	1,000秒×10
CS004	ボード特性確認	1, 10	1,000秒×2
CS005	発熱安定性確認	1	10,000秒
CS006	発熱安定性確認	10	10,000秒
CS007	温度上下確認	1, 10	1,000秒×2
CS129	クロック (昇順) 発生	0	10秒×13 ^{†††}
CS130	クロック (昇順) 発生	0	100秒×13 ^{†††}
CS131	クロック (降順) 発生	0	100秒×13 ^{†††}

[†] 全1,000ROHを各10サンプリング, ^{††} 1 Hz, 10 Hz, 100 Hz, 1 KHz,
^{†††} min. 1 Hz, max.100 MHz.

4.5.2 以降で説明する「チップ温度」は5回測定した T_C の平均値を使用している。図4-8は T_A が1,000秒安定した値を示す期間中の T_C を示す。2.2で述べたように、温度サイクル試験では、10分（600秒）間の温度維持が求められる。ここで、 T_C は数LSBのノイズを伴うことが確認できる。4.4.4で述べたように、オンチップ温度センサはADCと組み合わせたミックスドシグナル回路XADCとして実装されている[31]。また、複数のチップ内個別熱源から熱的に合成した温度が、チップ温度センサの測定対象として期待される。このため、ミックスドシグナル回路が高速なデジタル回路に近接して配置され、耐ノイズ設計が十分ではないと考えられる。

このノイズはシステム要求次第で、一段と低減させる必要がある。そこで、多項単純移動平均フィルタのノイズ低減効果を確認した。図4-8に示す信号ノイズ幅は5項移動平均でおおよそ1/2に、50項移動平均でおおよそ1/10に低減した。なお、本章では T_A が $25.0 \pm 1^\circ\text{C}$ に周囲温度が制御されている条件で T_B , T_C , T_E を測定した。

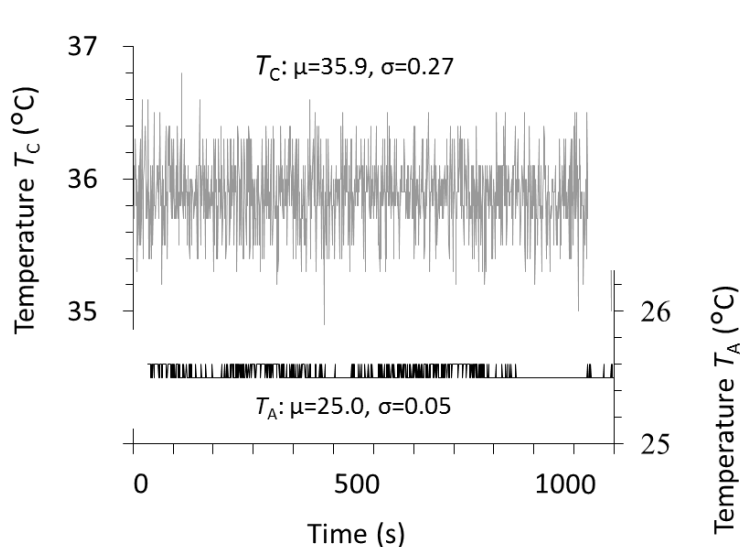


図4-8 室温におけるチップ温度の測定結果
チップ温度（上）とプラットフォーム周囲温度（下）

4.5.2 参照 ROH モジュール

4.2 で述べたように、RO の発振精度、及び安定性は過度に期待できないことが知られている。まずは、ROH モジュールを 100 組実装し、発振周波数とジッタの程度を確認した。

図 4-9 に発振周波数を 5 MHz 毎に分割し、それぞれの分割区間別の ROH モジュール数を示す。図 4-9 から左右に広がり中央付近のモジュール数が多い分布を確認できる。また、分布図に重ねた理論的な正規分布曲線と形似であることがわかる。このため、分布の母平均 μ を 100 組の代表値として扱う。分布におけるばらつきは、ROH ごとの配線ブロックの構成と伝播遅延の関係によるものと考えられる。ここで、 μ とする 207 MHz に近い発振周波数を出力する ROH モジュールを、4.5.3 以降の実験で参照モジュールとして扱う。なお、この分布にて 2σ は 50.6 MHz となり、同じ回路合成であっても配置・配線の差が発振周波数にわずかながらも影響することがわかる。また、無作為に選択した 8 モジュールでは発振周期の 0.01% オーダのジッタを確認した。

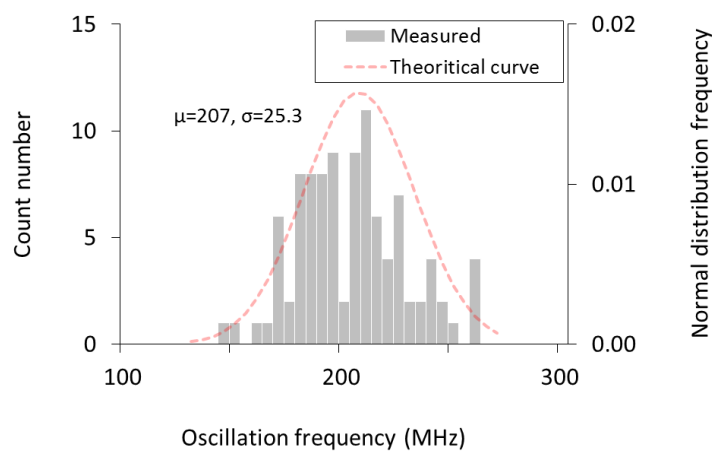


図 4-9 ROH モジュールの発振周波数分布
(GD 段数=8, 100 モジュール)

4.5.3 発振周波数の制御

つぎに、参照モジュールの遅延段数が発振周波数に与える影響について説明する。図 4-10 は参照モジュールについて測定した発振周波数と遅延段数の関係、及び発振周波数から求めた遅延時間と遅延段数の関係を示す。図 4-10 から遅延段数に対して遅延時間はほぼ一次線形とみなせる。平均すると、単一 GD 段あたりの遅延時間は 270 ps で、遅延段数 1 の発振周波数は 570 MHz である。一方で、図 4-10 における遅延時間の一次線形に対する複数の不連続点は、4.3.2 で述べたように遅延段となる GD が論理ブロックをまたがって実装されたとみなせる。

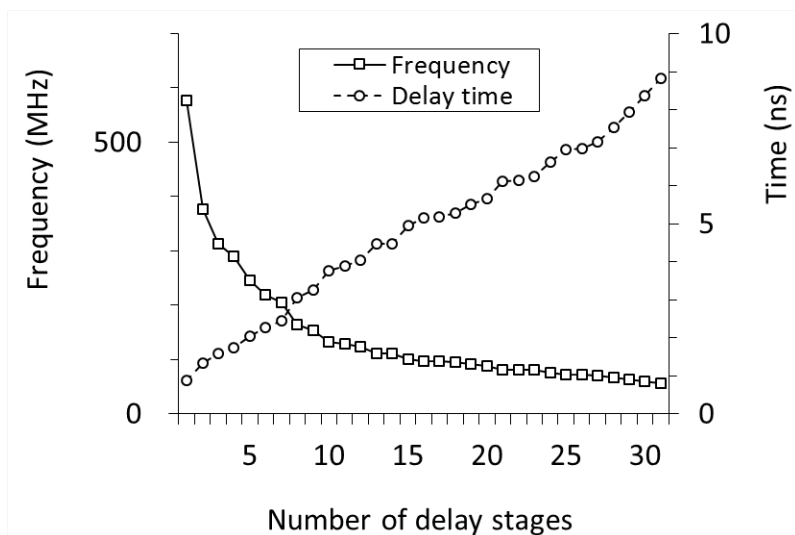


図 4-10 参照モジュールの発振周波数・遅延時間と遅延段数の関係 (GD 段数=32)

4.5.4 加熱の制御

ここでは、チップ温度を対象として推奨動作温度 85°Cを超える加熱の能否と、定格温度 125°Cへの到達見込みにより制御特性を評価する。そこで、ROH の3制御変数である (1) 遅延段数, (2) 駆動段数, (3) モジュール数の影響を測定した。なお、ここで示す T_c は FPGA に ROH の活性開始から 900 秒~1,000 秒の平均チップ温度である。

第一に、ROH モジュール遅延段数 1~8 に対する T_c の測定結果を表 4-7 と図 4-11 に示す。モジュール数の影響を検討するため、モジュールを 1 群、及び 10 群で活性化した。GD 段数は 16 であり、遅延段数 1~8 に対応して駆動段数は 15~8 となる。 T_c は図 4-10 に示した発振周波数に依存することがわかる。0 群はすべての ROH モジュールにとって待機状態であり、チップにとって放置冷却期間となる。遅延段数 2 で 10 群を活性化したモジュールは、チップを推奨動作温度以上の 86.7°C に加熱した。このことから、10 群未満のモジュールであっても推奨動作温度への加熱に余裕があると考えられる。

表4-7 チップ温度 T_c と遅延段数との関係 ($T_A=24.6^\circ\text{C}$)

Groups †	Number of delay stages							
	1	2	3	4	5	6	7	8
0	35.8 ††							
1	47.5	43.8	41.9	41.9	40.2	40.0	39.7	38.7
10	102.6	86.7	77.8	71.8	66.8	64.6	61.4	59.1

† number of GD steps: 16, †† average value.

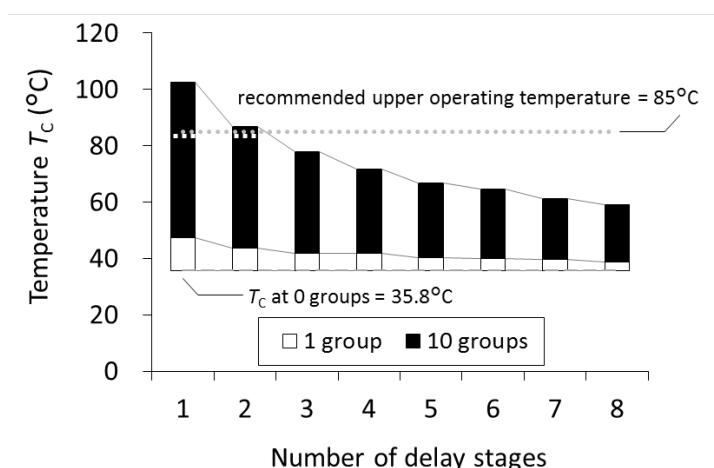


図 4-11 チップ温度 T_c と遅延段数の関係 (GD 段数=16, $T_A=24.6^\circ\text{C}$)

加えて、4.4.3 で述べた PU をターゲットとした温度特性の試験方法と試験結果を説明する。まず、CM に 32bit 長の Fibonacci 数列計算プログラムを書き込む [37]。つぎに、iMCM からプログラム実行を指示する。そして、IO が外部に通知する計算結果から PU の正常動作を確認した。室温で動作クロック 50 MHz での正常動作を確認した後、ROH を活性化しチップを加熱した。FPGA のチップ温度が推奨動作温度 85°C を超え 100°C に到達するまで、PU の正常動作を確認した。

第二に、いくつかの GD 段数（最小 1，最大 56）に対する T_C の測定結果を表 4-8 と図 4-12 に示す。ここでは、駆動段数の影響確認に集中するため遅延段数を 1 に固定した。したがって、GD 段数 1~N は駆動段数 0~N-1 に対応する。モジュール群数 10 の活性化により (1) 駆動段数 0 の基準チップ温度は 67.1°C に、(2) 15 以上の駆動段数でのチップ温度は推奨動作温度以上である 102.6°C に、(3) 駆動段数 31 では定格温度に極めて近い 123.7°C にチップが加熱されたことを示す（表 4-8）。31 を超える駆動段数では定格温度超えが確実とみなされるため、モジュール群数 9 で測定した相対値で示している（表 4-8 最下行の%表示）。駆動段数がチップ温度に影響するこれらの測定結果から、発振出力が伝播する配線路長がチップ加熱に影響を及ぼしている。

表4-8 チップ温度 T_C と GD 段数との関係 ($T_A=24.6^\circ\text{C}$)

Groups	Number of GD steps †							
	1	8	16	24	32	40	48	56
1	40.9	45.6	47.5	50.0	53.5	55.4	56.4	57.5
10	67.1	95.4	102.6	115.8	123.7	—	—	—
(%)	—	—	—	—	(100)	(106)	(108)	(118)

†=number of drive stages+1.

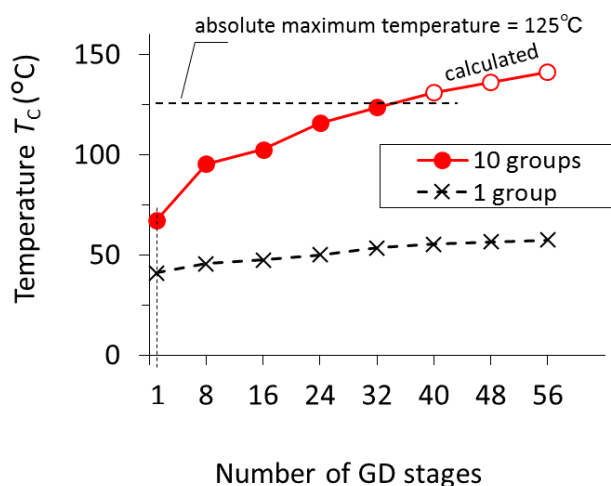


図 4-12 チップ温度 T_c と GD 段数の関係 (遅延段数=1, $T_A=24.6^\circ\text{C}$)

10 群の GD 段数 40 以上のチップ温度は、9 群の温度上昇率からの計算値

第三に、モジュール群数 1~10 に対する T_c の測定結果を図 4-13 に示す。さらに、モジュール群数 1~10 に対する T_c の時間変化率を測定した結果を表 4-9 に示す。変化率は次のように求める。(1) 測定対象のモジュール群数で開始チップ温度から 1,000 秒間チップを加熱、(2) 1,000 秒経過時における到達チップ温度と開始チップ温度との全体温度差を 100%とし、(3) 測定開始から経過時間までの経過温度差と全体温度差に対する割合を時間変化率とする。加熱開始から 100 秒程度のオーダーでチップ温度はゆるやかに上昇するが、モジュール群数にかかわらず時間比率に大きな差が認められないことが表 4-9 からわかる。たとえば、20%までは 50 秒、60%までは 100 秒、90%までは 200 秒を加熱に要している (図 4-14)。モジュール群数 10 では最初の 1 分 (60 秒) で $12^\circ\text{C}/\text{分}$ の速度でチップ温度を上昇させ、一般的な恒温槽のおよそ $3^\circ\text{C}/\text{分}$ と比較するとほぼ 4 倍の加熱能力があるといえる。図 4-13 が示す加熱開始からのゆるやかな温度上昇は、熱伝導率 10^3 オーダ差のあるチップとパッケージの熱結合が要因と考えられる。差分法を用いて、パッケージ材料となるエポキシ樹脂と同オーダーの熱伝導率を設定したシミュレーションでは、温度変化が 100 秒オーダーとなることを確認した [68]。なお、温度の上昇幅が 1°C 程度と小さい場合においても、ゆるやかなチップ温度上昇と同様に観察できた (図 3-8)。

一方、モジュール群あたりの加熱温度すなわち加熱効率は $8.1\sim 4.5^\circ\text{C}/\text{群数}$ と幅があることから、群数が大きいほど効率が小さくなる傾向が確認できた (表 4-9 最下行)。この傾向はモジュールの発振周波数に対する温度特性による影響と考えられる。したが

って、100°Cを超える高温への加熱には加熱効率を考慮した ROH の制御変数選択が必要となる。

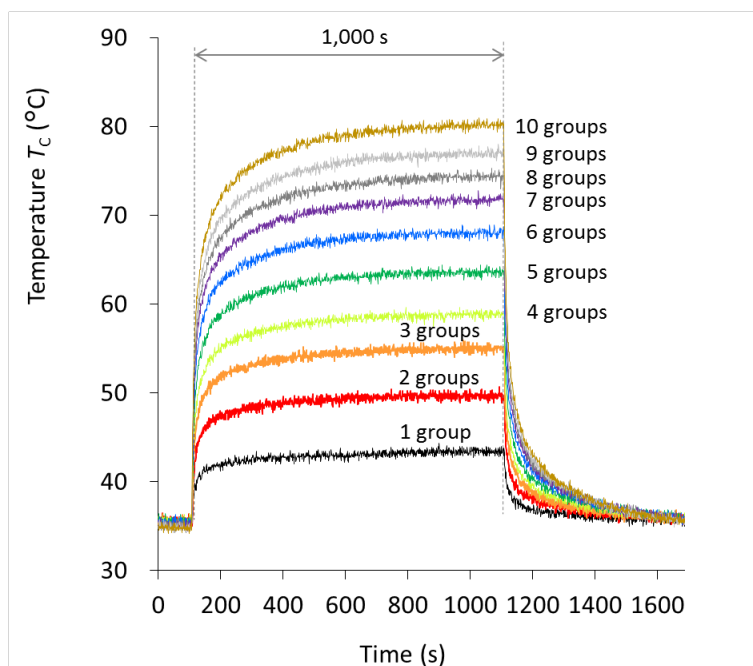


図 4-13 加熱温度変化と ROH モジュール数の関係

表4-9 チップ温度上昇の時間率とROH群数の関係 (%)
 (群数=1~10, GD段数=16, 遅延段数=1, $T_A=24.3^{\circ}\text{C}$)

Elapsed time (s)	Number of module groups										AVG.
	1	2	3	4	5	6	7	8	9	10	
0	0	0	0	0	0	0	0	0	0	0	0
50	21	22	21	21	21	21	20	20	19	19	21
60	29	29	29	28	28	28	27	27	27	27	28
100	59	60	60	60	59	59	59	58	58	58	59
200	85	87	87	87	87	87	87	86	86	86	86
300	90	92	92	92	92	92	92	92	92	92	92
400	93	95	95	95	95	96	95	95	95	95	95
500	94	96	97	97	97	97	97	97	97	97	97
600	95	98	98	98	98	98	98	98	98	98	98
700	97	98	99	99	99	99	99	99	99	99	99
800	99	99	99	99	100	100	100	99	99	99	99
900	99	100	100	100	100	100	100	100	100	100	100
1,000	100	100	100	100	100	100	100	100	100	100	100
End† (°C)	43.4	49.7	55.0	58.9	63.6	68.0	71.7	74.4	77.0	80.2	
(°C/groups)	8.1	7.2	6.6	5.9	5.7	5.5	5.2	4.9	4.6	4.5	

† start temperature= 35.5°C for all groups.

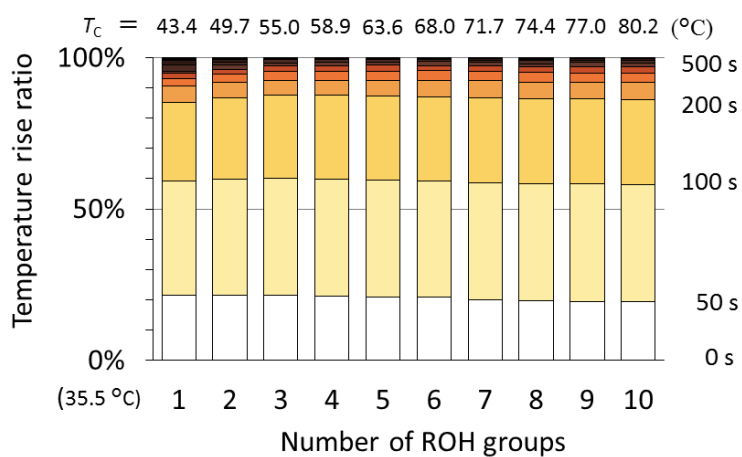


図 4-14 チップ温度の上昇率

各系列の下からの要素は、
 加熱開始した 0 s から 50 s, 100 s, 200 s, (300 s, 400 s,) 500 s を
 経過したことを示している

4.5.5 加熱の均一性

4.2 で述べたように、高温の空気を槽内に充満させる恒温槽は、ボードやデバイス、デバイス内の回路種類・個数・位置によらず全体を均一に加熱する方式といえる。ここでは、恒温槽の実用的代替として ROH を高温テストに供するために、実験結果を考慮しつつチップ加熱の均一性を説明する。

まず、チップを 100°C まで加熱した FPGA デバイス付近のボード温度を、赤外線画像（FLIR Systems 社の FLIR ONE Pro にて撮影）で図 4-15 に示す。次に、簡易恒温槽にボードを設置し、槽内温度を 80°C に加熱した FPGA デバイス付近のボード温度を赤外線画像で図 4-16 に示す。

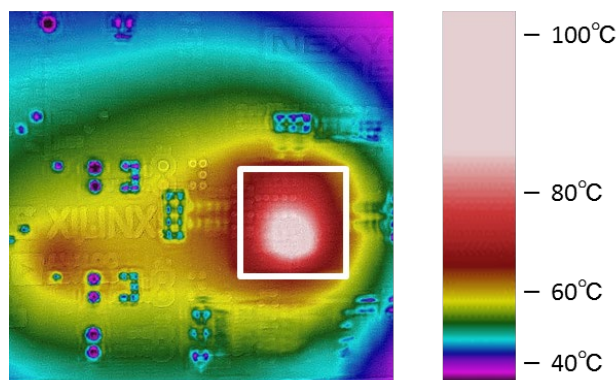


図 4-15 ROH により加熱された
FPGA デバイス付近の IR 画像

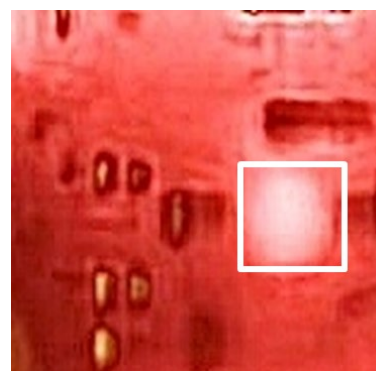


図 4-16 外部加熱時の IR 画像

図 4-17 に簡易恒温槽の構成を、そして図 4-18 に構造の外観を示す。図 4-15 では白色枠の位置に FPGA デバイスが実装されており、次の温度分布の特徴がわかる。(1) 最も温度の高い領域はチップ中央から外れている、(2) チップ形状と形似の矩形ではない、(3) チップ外周部では最高温度から 10°C ほどの温度差である。(2)、(3) は熱伝導率 10^3 オーダ差のあるチップとパッケージの熱結合が要因と考えられる [24]。

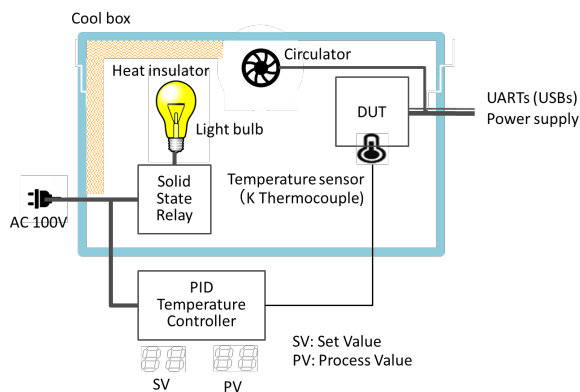


図 4-17 簡易恒温槽の構成

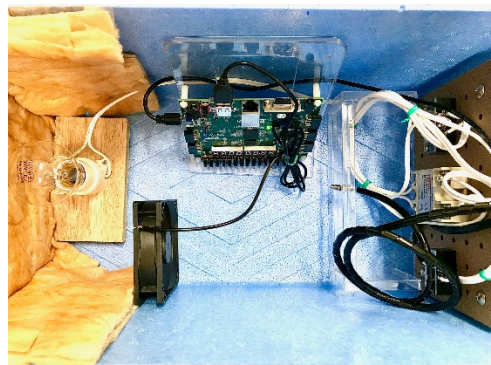


図 4-18 簡易恒温槽の構造外観

そこで、ボードの FPGA デバイス付近の X 線透過画像（東芝の TOSCANER-32250 μ hd にて撮影）から、パッケージ内のチップ位置を確認し、そして大きさを測定した。この FPGA デバイスでは外形 15 mm のパッケージ (wire bond CSG : Chip Size BGA) に、 $6.93\text{ mm} \times 10.05\text{ mm}$ のチップが封入されていた (図 4-19)。さらに、パッケージとチップ位置を発熱時の IR 画像と重ねると、最も温度の高い領域はチップ外形内に留まっていることが分かる (図 4-20)。

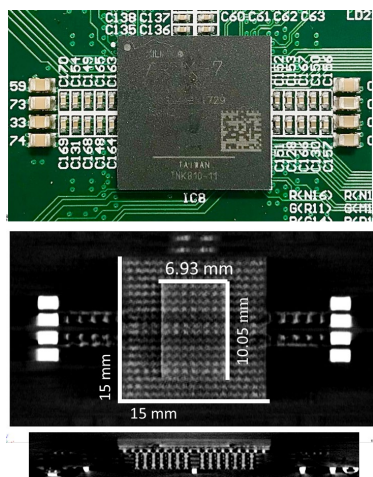


図 4-19 FPGA デバイスの X 線透視画像
(上) 可視画像, (中) 水平面, (下) 断面

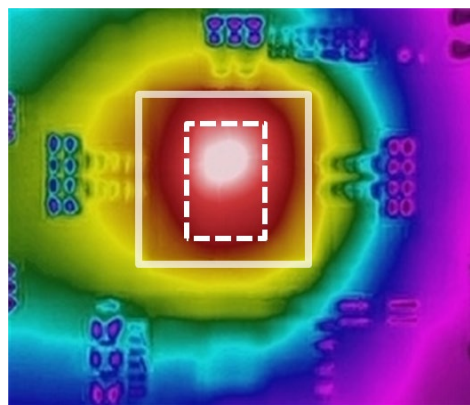
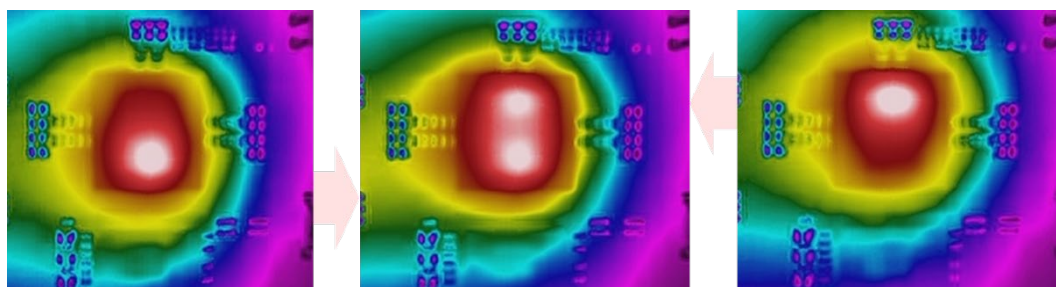


図 4-20 FPGA デバイスの IR 画像
(パッケージを実線, チップを点線で表示)

これらの特徴から、目標温度とテスト対象回路の温度差がテストに影響する可能性として (1) テスト対象回路と ROH の位置関係、(2) オンチップ温度センサの位置が考えられる。(2) は FPGA デバイス毎に固定されており、設計者が移動できない制約がある。そこで、テストの目的によってはオンチップ温度センサの位置を考慮して、テスト対象回路と ROH を配置する必要がある。図 4-15 からは温度分布の差は 10°Cほどある。しかしながら、恒温槽を使用する場合と比較して、ROH による加熱の利点（たとえば、広い場所・大きな電力を必要としないことや、利用者が運用環境で試験が可能であることなど）は、多くの領域で提案 ROH が適用可能であることを示している。

一例として、特定の機能・構造の回路配置を狙った加熱がある。2組の 1,000 モジュール ROH である ROH_1 と ROH_2 により加熱した、FPGA デバイス付近のボード温度を赤外線画像で図 4-21 に示す。図 4-21 (a) はチップ下方の ROH_1 により加熱した画像であり、図 4-21 (c) はチップ上方の ROH_2 により加熱した画像である。図 4-21 (b) は ROH_1 及び ROH_2 が同時に加熱した画像である。図 4-21 (b) から、2つの加熱領域が独立して加熱する様子が確認できる。加熱対象回路の配置を把握しておくことにより、特定の回路を選択して部分的・集中的な加熱への応用が期待できる。



(a) ROH_1 による (b) ROH_1 + ROH_2 による (c) ROH_2 による

図 4-21 複数の ROH により加熱された FPGA デバイス付近の IR 画像

4.5.6 実験結果と考察

ここでは、4.4.6 で述べた実装結果と、4.5.1～4.5.5 で述べた実験結果から、提案 ROH と既報手法の比較を示して、比較結果を考察する。表 4-10 は RO を FPGA チップ加熱に関連した実装と実験の結果を、いくつかの変数で比較している。ここで、ROH の消費電力はボード全体の消費電力 P_{in} から算出した参考値 ΔP_{in} である。 P_{in} には ROH 以外の FPGA 実装回路や FPGA 以外のボード搭載部品が要する消費電力を含む。ROH 活性化に要する P_{in} のみを表す ΔP_{in} は、ROH モジュール非活性化時に $P_{in_ROH_off}$ とし、さらにモジュール 10 群での活性化時に $P_{in_ROH_on}$ として式(4-2)で示される。表 4-10 に、BH 及び SHE も加熱に要する消費電力のみを示している。

$$\Delta P_{in} = P_{in_ROH_on} - P_{in_ROH_off} \quad (4-2)$$

正確な消費電力の比較には V_{CCINT} を発生する DC-DC コンバータにおける電力変換効率の補正が必要になる。式(4-2)では効率を 100%としている。また、公平な比較にはそれぞれで実装した FPGA デバイス差の理解が必要である。どの FPGA も Xilinx 社の製品であり、基本構造だけでなく開発環境にも継承性がある。2 組の 6 入力 LUT×4 (SLICE) が一つの論理ブロック (CLB) となる。ROH を実装した 28nm プロセス低消費電力の Artix-7 (XC7A100T-1CSG324C) は、ヒートシンクなしでボード (Nexys4 DDR) に搭載した。一方、BH と SHE を実装した 65nm プロセス高速・高性能の Virtex-5 (XC5VLX110T-1FFG1136C) は、ヒートシンクを装着してボード (XUPV5-LX110T) に搭載したことが考えられる。これら FPGA デバイスと搭載方法の差により、周囲の外気とジャンクション間の実質的な熱抵抗 θ_{jc} の考慮が必要である。なお、Artix-7 のスタティック電力、及びダイナミック電力は Virtex-5 と比して、50%以上低減されている [69], [70]。

表 4-10 の比較結果の特徴と推定要因を次のように整理する。(1) 加熱したチップ温度の差は、それぞれのデバイスの推奨動作温度 (ROH は 85°C, BH と SHE で 125°C) にて加熱を抑制したためで加熱能力の差ではない、(2) 高速・高性能 FPGA に実装した SHE と ROH の両方で RO の最大発振周波数が同程度なのは、ROH が 3 世代 (65nm → 45nm → 28nm) に渡るプロセス技術進歩の恩恵を受けた、(3) 占有 LUT 数は加熱温度を考慮すると、BH と SHE に対する ROH の増分は明確に 50%を超えた、(4) BH と SHE が目標温度までチップ加熱に要した消費電力に対して同程度の ROH では θ_{jc} の他に、加熱器を構成する個別熱源で分散する発熱タイミングも影響する、(5) BH と SHE に対する ROH の RO 数比率は、Wiring block の未使用 / 使用の差に起因する。

表4-10 提案ROHと関連研究の緒元比較

Parameters	This work	Related works	
		[Waber (2014)]	[Amouri (2015)]
Basic heater FPGA device (Xilinx)	ROH † Artix-7	BH Virtex-5	SHE Virtex-5
Heating temperature (°C)	124	> 150	> 120
Power consumption (W)	8.3 ††	10.8	14 ††
Number of heaters	1,000	1,480	1,440
Number of ROs	1,000	11,840 †††	10,800 †††
LUTs (6 inputs)	17,829	11,840 ††	11,520
Oscillation frequency (MHz)	570	N/A	500 ††

† number of GD stages: 16, †† min. : 1,480,

††† by simulation, †††† by calculation.

ここまでの実験結果は、低消費電力 FPGA に実装した ROH が既報の高速・高性能 FPGA と同程度の加熱特性を得たことを示す。さらに、前述の推定を総合すると、ROH は高速・高性能 FPGA (たとえば、Xilinx 社の Virtex-7 や UltraScale 以後の世代) では、より占有 LUT 数の少ない加熱器の実現が期待できる。

4.6 チップ温度制御への応用

ここでは、4.4.3 で述べた熱制御器 (TC) の一例を示し、FPGA に実装した TC の加熱と温度維持の能力を実験結果から説明する。4.5.5 で述べたデバイスを構成する複数材質の熱伝導率差が、チップ温度を秒から分オーダーで変化させる。熱制御を実環境で運用するには加熱と温度維持に要する消費電力の最小化が望まれる。有効な制御方式には目標チップ温度までの加熱に電力を加え、その後加熱を緩やかにするよう電力供給を低下させる比例制御 (P 制御) が考えられる。

図 4-22 に TC の機能構成を示す。ROH モジュール群による加熱の影響は、合成熱抵抗容量 (thermal CR) を経てチップ温度センサ (CTS) が温度データに変換した測定結果から確認できる。ノイズフィルタ (noise filter) は温度データの変換ノイズを低減する。プロフィール記憶装置 (profile memory) には ROH モジュール活性化数に対する加熱実績情報を記録しておく。記録した実績情報と測定した温度データ、活性化したモジュール数実績を参考に次の制御単位時間における活性化モジュール数が指定される (module select による)。観測した温度と設定した温度 (target temperature) 差を入力とする簡単な P 制御により、目標チップ温度への加熱を制御した。4.5.4 で述べたように、加熱温度変化は 100 秒程度の単位であり制御単位時間には 1 秒を選択した。また、ROH モジュール活性化数に 1 と 5, 10 群の 3 種を選択した。TC はまずチップ温度が安定していることを確認するために 100 秒の待ち時間を確保して、安定したチップ温度から加熱を開始した。

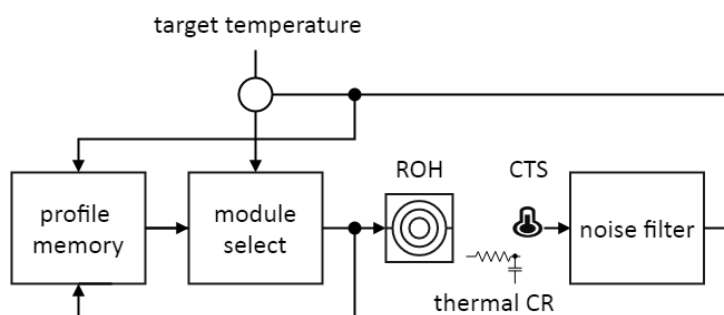


図 4-22 熱制御器 (TC) の機能構成

図 4-23 に目標チップ温度を 65°Cとしたチップ温度と電源電流の時間変化を示す (Keysight Technologies 社の 34465A にて電源電流を測定)。4.5.4 で述べた対目標チップ温度 90%までの加熱時間 200 秒を, TC が制御した提案 ROH は加熱開始から 2 分の 1 ほどの時間に短縮した。目標温度への到達までは 10 群の, 到達後は 5 群の ROH モジュールが活性化される。チップ温度が目標温度を下回りはじめると, 1 秒ないし 2 秒期間 10 群の ROH モジュールを再活性化することで, 電源電流がパルス状に変化している。

図 4-24 には目標チップ温度を 75°Cとしたチップ温度の時間変化を示す。図 4-24 には目標チップ温度を 65°Cとしたチップ温度は比較のために示す。目標チップ温度を 75°Cとした場合は, 目標温度に到達する間近で温度上昇が緩やかになるものの, 到達後は安定することを示している。

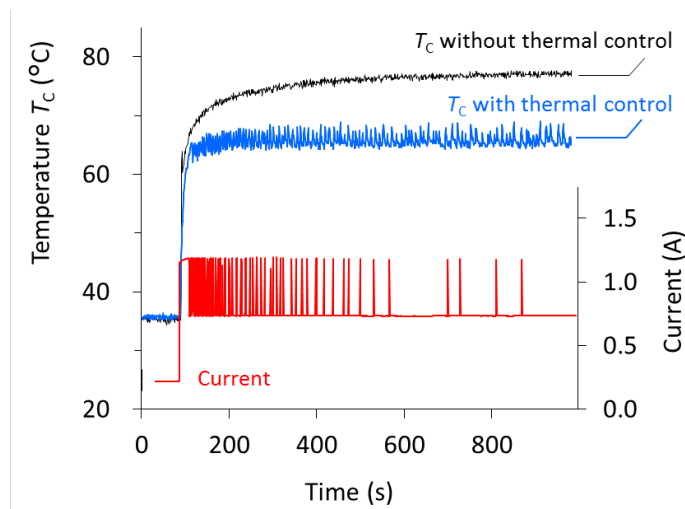


図 4-23 熱制御器 (TC) により制御されたチップ温度と電源電流 ($T_A=24.6^{\circ}\text{C}$)

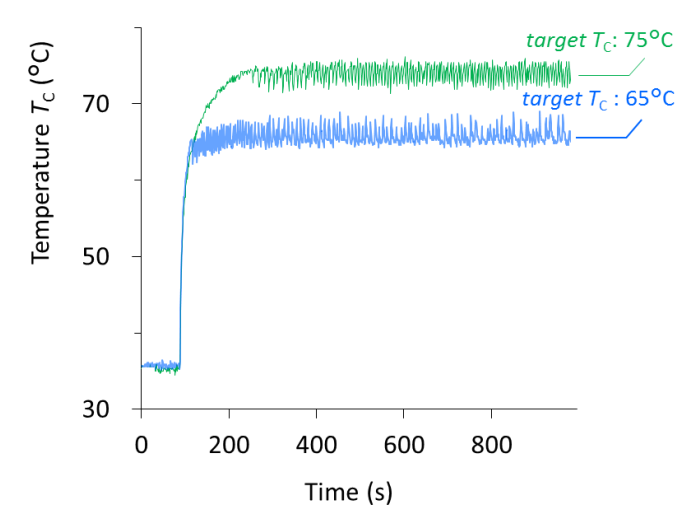


図 4-24 熱制御器 (TC) により制御されたチップ温度のターゲット温度による差

4.7 おわりに

本章では FPGA に設計者が実装した回路プログラム結果の温度特性を試験する技術として、拡張駆動リングオシレータ加熱器を提案した。従来は駆動対象に考慮されなかったフィードバック信号を、提案したリングオシレータ加熱器では拡張駆動負荷として加熱に利用する。リング内で遅延時間と駆動負荷を同時に選択することにより、試験時のチップ加熱設定用変数の選択幅を広げた。更に、推奨温度を超えるチップ温度実現と定格温度に達するチップ温度傾向を、FPGA への実験評価により実証した。実験には 28nm プロセスを採用した市販 FPGA デバイスと市販 FPGA プラットホームを使用することで、実験結果への再現性を確保した。およそ 18,000 の LUT を用いて推奨動作温度 85°C まで、またおよそ 34,000 の LUT を用いて定格温度 125°C までチップを加熱した。従来とは異なる低消費電力 FPGA への実装においても、同程度の加熱特性が示された。

今後の課題には、まずオンチップ温度に関する変換ノイズ低減、つぎに温度特性を組み込んだ加熱効率のモデル化があげられる。そして、実装実験における対象 FPGA の拡大並びに MPU の温度特性測定と評価から動作限界となるチップ温度の判定を行う。

第5章 FPGA 搭載プロセッサ向け機械語モニタの ハードウェア実装

5.1 はじめに

インダストリー4.0 (Industry 4.0 : I4) が実現を狙うスマートファクトリ (smart factory) を実現するために、製造連鎖 (engineering chain) と供給連鎖 (supply chain) のネットワーク化が欠かせないと言われている。このネットワークにつながる先進的な技術として、モノのインターネット (Internet of Things : IoT) と産業用ロボットが期待される。これらの技術における代表的な課題に、プロセッサハードウェアと組み込みソフトウェアを搭載する組み込みシステムのカスタマイズ化がある。以下、2つの例を示す。

(1) IoT の中核技術のひとつであるセンサネットワーク向け応用機器では、処理性能・電力効率の両立を要求する組み込みシステムが増えつつある。これらの要求に最適に応じる ISA を実装する新規プロセッサ採用が選択肢となる。

(2) 産業用ロボットの関節モジュールは、サーボメカニズムとともに検知・判定・制御などの機能を管理する組み込みシステムを搭載する。各モジュールに割り当てられた役割によって、組み込みシステムに要求される機能や処理性能は異なる。様々なモジュールをサポートする組み込みシステムでは、標準的な ISA への追加・変更・削除を伴うカスタマイズされたプロセッサが必要となる。

新規プロセッサの設計検証段階、及びカスタマイズされたプロセッサの設計検証段階では、固有 ISA のサポートに必要な不可欠な機械語モニタ (Machine Code Monitor : MCM) が提供できない。MCM はファームウェアやシステムソフトウェアの一つとしてセルフ (self environment) 環境でのプログラムとして提供されてきた。MCM のオブジェクトコード作成には、クロス環境 (cross environment) におけるソフトウェア開発ツールが必要である。開発ツールの追加や改版を制限すれば、提供時期とともに固有 ISA のユニークさは限定されたものになる。

本章では、プログラムとしてではなく回路として実装する MCM について述べる [66], [71], [72]。プログラミング言語ではなく、ハードウェア記述言語 (Hardware Description Language : HDL) で記述し FPGA にプロセッサとともに実装する。この実装により、利用者の要求に応じてモニタ機能を再構成し、開発ツールの整備状況に依存せずに機械語モニタの機能を提供する。MCM の利用者は主にプロセッサ開発者とシステムソフトウェア開発者である。

以下、本章は次のように構成される。まず、5.2 でプロセッサの開発環境と従来の機械語モニタの役割を議論する。5.3 では、提案する機械語モニタの構成、動作、及び設計手法を紹介する。5.4 では、モニタ機能の単体実装実験を説明するとともに、評価の結果を議論する。5.5 では、モニタ機能にターゲットプロセッサを統合する際の要件と期待できる貢献について議論するとともに、実際にプロセッサの設計経験者による機械語モニタのニーズをヒアリングした結果を紹介する。5.6 では、ターゲットプロセッサの紹介と設計方針を議論する。5.7 では、統合実装実験を説明するとともに、評価の結果を議論する。そして5.8では、提案した機械語モニタの更なる展開について議論する。最後に、5.9 で本章を締めくくる。

5.2 プロセッサの開発環境

FPGA に組み込まれるプロセッサ

外部の自然環境が大きく変化する水中や宇宙空間での運用とともに、5.1 で述べた組込みシステムへの FPGA の適用検討が広がっている [73], [74], [75], [76], [77], [78]. 一方, 順序制御を多用する組込みシステムや, ソフトウェア再利用を重視する高信頼システムではプロセッサの選択が主流である. このためメーカ独自のプロセッサや他社の IP コアを組み込んだ FPGA の商用化も進んでいる [79], [80]. なお, メーカによってはプロセッサを組み込んだ FPGA を SoC と分類する場合があるが, 本研究はすべて FPGA として分類する [81]. さらに, 標準 ISA を追加・変更・削除する固有 ISA を実装するプロセッサの組込みも研究されている [48]. この種の組込みプロセッサのソフトウェア開発ツール (アセンブラ・コンパイラ・シミュレータ・デバッガなど) はプロセッサとの同時公開が望まれている. これらの開発ツールのうち, 教育や非営利研究などの特別な利用者向けに提供されているのはモニタ機能のみである [82]. モニタ機能はホスト PC 上のクロス環境で実行される. そのため, ホスト PC と FPGA 上のプロセッサとの同期が保証できず, モニタ機能の処理時間が変動する.

本章では, RISC-V ベースの ISA を搭載したプロセッサに MCM を統合して合成した統合 MCM (Integrated MCM: iMCM) を提案する. 図 5-1 に MCM, 図 5-2 に iMCM の開発フローの概要を示す. iMCM は, 主に FPGA の LUT やメモリなどのリソースが限られている組込みシステムへの応用を想定している.

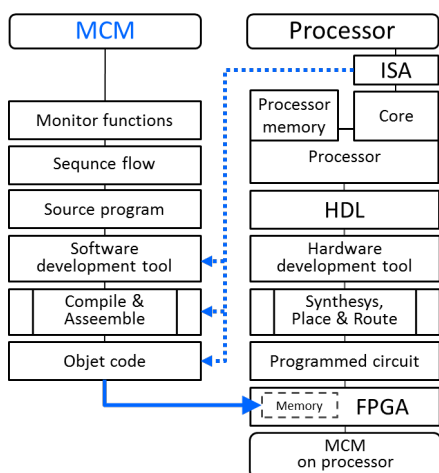


図 5-1 従来 MCM の実装フロー

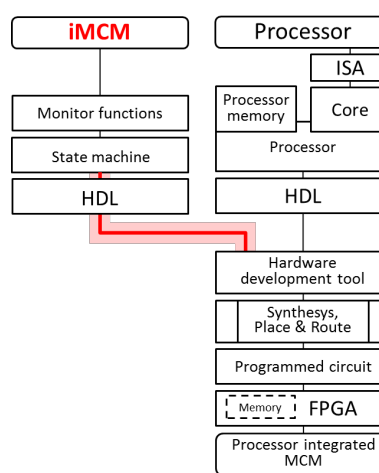


図 5-2 提案 iMCM の実装フロー

プロセッサの開発環境

図 5-3 に FPGA に搭載するプロセッサ用開発環境の構成例を示す。FPGA 上のプログラマブル部はプロセッサ，周辺回路とハード IP を含み，FPGA プラットホーム上の JTAG を介してクロス環境と接続する。クロス環境上のハードウェア開発ツールは HDL 記述から論理合成，及び配置配線を経た回路プログラムをビットストリームとして FPGA にダウンロードする。クロス環境にはソフトウェア開発ツール一式が搭載され，ネイティブ環境には FPGA が単独で動作する際に必要なシステムソフトウェアとランタイムライブラリが搭載される。

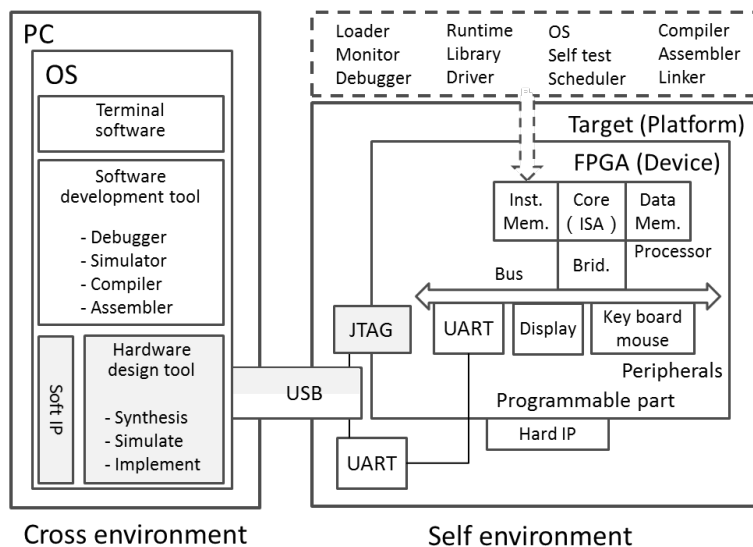


図 5-3 FPGA 実装プロセッサの開発環境の構成

機械語モニタの役割

プロセッサの設計検証フェーズでは、プログラムの実行対象となる命令メモリとデータ格納用のデータメモリからなるメモリ群（プロセッサメモリ）の管理が特に重要である。プロセッサメモリ管理の第一歩は機械語（machine code）の転送である。転送にはクロス環境からプロセッサメモリへと、プロセッサメモリからクロス環境方向への2種類がある。利用者が試行錯誤しながら機械語を確認し設定・変更管理する開発ツールが機械語モニタである。機械語モニタは命令メモリに格納され、利用者が操作と表示を繰り返すモニタ機能を提供する。また、プラットフォーム上の UART は機械語モニタとクロス環境上の端末ソフトウェアをシリアル通信で接続する。

代表的なモニタ機能には、(1) メモリ内容の表示と変更、(2) プログラムの実行開始と中断、(3) オブジェクトコードのダウンロードなどがある [83], [84], [85], [86], [87]。プロセッサの検証と修正のサイクルを効率よく繰り返すためには、すべての命令が固有 ISA を正しく実装したプロセッサ上での動作が前提となる。したがってプロセッサへの固有 ISA 実装が完了するまで機械語モニタは正しいモニタ機能を提供し得ない。

iMCM が提供する 7 つのモニタ機能は、FPGA への実装経験を持つ 6 名のプロセッサ設計者からのヒアリング結果をもとに、以下の優先順位で選定した。(1) プロセッサメモリの読出しと書換え、(2) 命令メモリへの HEX ファイルのロード、(3) 指定アドレスからの実行開始、(4) レジスタファイルの読出しと書換え、(5) プロセッサメモリとレジスタファイルの初期化、(6) シングルステップ実行と可視速度での実行、(7) プログラムトレースでのプロセッサ資源と状態の表示、の 7 つのモニタ機能を優先的に選定した。

5.3 提案する機械語モニタ

ここでは、プロセッサの設計品質レベルと、ソフトウェア開発環境の整備レベルに依存せず、HDL 記述から FPGA へ回路として実装する機械語モニタについて構成・動作・設計手法を述べる。

5.3.1 構成概要

図 5-4 に iMCM のブロック構成を示す。図 5-4 の最上部は UART を介したクロス環境上の端末ソフトウェアとのインタフェースを、最下部はプロセッサメモリとレジスタファイルのインタフェースを示す。また、図 5-4 の左部分が操作関連処理、そして右部分が表示関連処理をそれぞれ示している。UART に接続する文字列 1 行分（最大 256 B）のバッファ Command buffer を配置し、iMCM の動作と UART のシリアル通信速度のギャップを埋める。

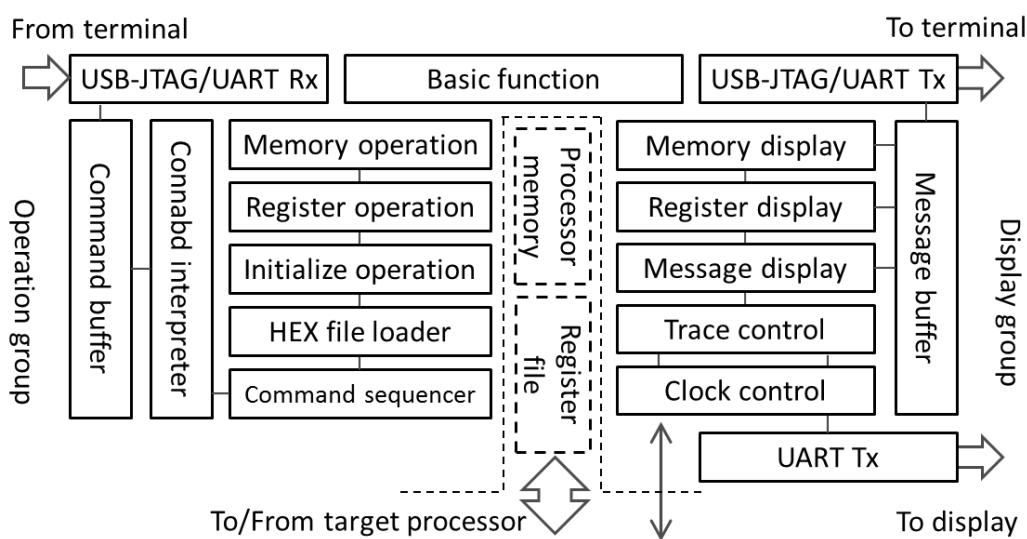


図 5-4 提案 iMCM のブロック構成

5.3.2 動作概要

図 5-4 に示した Command sequencer 処理は、利用者からの 1 行分の文字入力を Command interpreter 処理が UART を介して検出したときに起動される。起動直後に、1 行分のコマンド文字をコマンド機能として特定し、各コマンドに固有のパラメータを取得する。図 5-4 には、上述した 7 つのモニタ機能に対応する処理グループが含まれている。得られたコマンド文字に応じて、表示と操作のための処理が開始される(表 5-1)。

表5-1 モニタ機能と処理の関係

優先順位	モニタ機能	処理	
高 ↑	1	プロセッサメモリの読出しと書換え	メモリ表示 メモリ操作
	2	命令メモリへの HEX ファイルのロード	HEX ファイルローダ
	3	指定アドレスからの実行開始	トレース制御
	4	レジスタファイルの読出しと書換え	レジスタ表示 レジスタ操作
↓ 低	5	プロセッサメモリとレジスタファイルの初期化	初期化操作
	6	シングルステップ実行と可視速度での実行	クロック制御
	7	プログラムトレースでの内部資源と状態の表示	トレース制御

5.3.3 設計手法の概要

処理の実行とコマンド文字識別処理は独立して行われるため、プログラムに要求されるリアルタイム性の損失はない。コマンド文字識別処理では、HDL の case 文を使用し、1クロックで複数の比較と複数の分岐を同時に処理する。このように、iMCM はコマンド数に依存しないリアルタイム性能を実現している。図 5-4 に示した処理ブロックやバッファはすべて HDL で記述し、論理合成ツールと配置ツール・配線ツールを用いて各処理の回路として実装した。HDL の記述により、複数の処理を同時に動作させる回路構成を実装できた。同時動作はシーケンスフロー (sequence flow) ベースのプログラム記述では実現できない。そのため、プログラム記述から HDL 記述への変換では、順序処理や共通変数の一貫性を確保しなければならない。

この一貫性確保の課題を、iMCM ではステートマシン (以下 SM) の組み合わせによる逐次処理により解決した。まず、シーケンスフローからこれらのモニタ機能のイベントを抽出する。次の設計段階にて、イベント駆動処理を SM として変換した。

ステートマシンの設計

処理 X 用の SM の機能モデルを図 5-5 に示す。SM は処理要求で起動され、処理の受け付け確認を要求元に通知する。SM 内のステートは外部に SM の進行状況を通知するとともに、モニタの他部分からの状態信号によりステートの遷移に反映させる。特定のステートまで遷移すると処理 Y 用の SM の起動を要求する。この時、処理 Y 用の SM が処理中であれば要求を保留する。このように複数の SM で処理順序を確定できる。

ステートの基本的な遷移を図 5-6 に示す。最初のステート ($ms=0$) はアイドル状態で、req 信号のアサート (assert) により処理の起動を待ち合わせる。起動が要求されると 2 番目のステート ($ms=1$) に遷移し、処理要求の受け付け確認を busy 信号のアサートで外部に通知するとともに処理の実体となる動作を開始する。処理の実体 ($ms=2\sim n-2$) はプログラム実行と同様に、順次的にあるいは分岐または繰返して変数へ代入する。変数には処理固有のローカル変数と複数の処理にまたがる共通変数がある。共通変数への代入は SM 外部で行い、マルチドライブの発生を回避する。

処理の実体への対応がなくなった状態 ($ms = n-1$) はウェイト状態であり, end 信号で SM の終了を待ち合わせる. 最後の状態 ($ms = n$) では使用した変数をクリーンアップするとともに処理の完了確認を $busy$ 信号のネゲート (\overline{busy}) で外部に通知する.

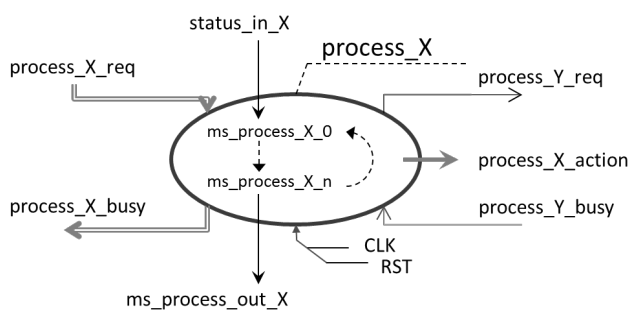


図 5-5 実装した SM の機能モデル

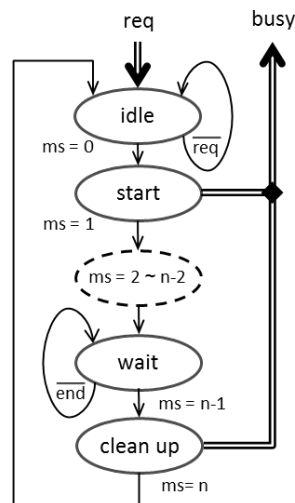


図 5-6 基本状態遷移図

プロセッサとのインタフェース

iMCM とターゲットプロセッサのインタフェースは局所的であり，大別して，(1) プロセッサメモリとレジスタファイルの入出力ポートの追加，(2) 動作クロックの再構成，(3) 割込み要求・受付の再構成がある．このインタフェースはプロセッサの設計後でも一部の機能追加と信号の接続・切断による再構成で実装できる．

図 5-7 にプロセッサメモリの入出力ポートを追加するインタフェース例を示す．モニタリングの対象とするレジスタファイルも同一方法でインタフェースを実装する．

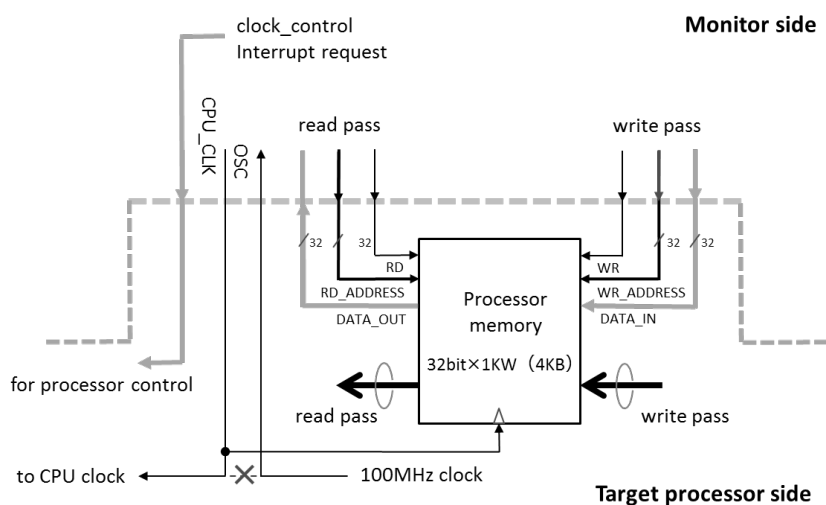


図 5-7 プロセッサメモリのインタフェース

5.4 単体実装実験と評価

5.4.1 実験プラットフォーム

表 5-2 に本章で使用した実装実験環境一式を示す。今後消費エネルギーと許容エネルギーの最適化の評価継続を視野に入れ、低消費電力動作の FPGA が搭載済みである開発評価システム一式(以後、プラットフォームと呼ぶ)として Digilent 社の NEXYS4 DDR を選択した [88]。プラットフォームに FPGA デバイスとして搭載される Xilinx 社の 7 シリーズ製品群のうち Artix-7 (XC7A100T-1CSG324) はロジックセル (歴史的には 2 入力 NAND ゲート) 換算 101,440 (LUT 換算では、さらに 1.6 で除した 63,400) である。この規模は同社 7 シリーズ製品全体の最小デバイスである XC7S6 の 6,000 から最大デバイスである XC7V2000T の 1,954,560 の範囲で中規模であり (LUT 換算ではそれぞれ 3,750 と 1,221,600 に相当する)、今後小規模システムあるいは大規模システムいずれの方向への移行も容易であると考えられる [46], [45]。

表5-2 実験環境

ワークステーション (PC)	CPU: Xeon 3.5 GHz, メモリ: 16 GB
OS	Windows 7 professional (64bit)
FPGA 開発ツール	Vivado 2016.2 (Xilinx 社)
HDL	Verilog HDL
端末ソフトウェア	RLogin (Culti)
ターゲット FPGA デバイス	Artix-7 (Xilinx 社)
FPGA プラットホーム	Nexys4 DDR (Digilent 社)
デジタルマルチメータ	34465A (Keysight Technologies 社)
IR カメラ	FLIR ONE (FLIR 社)

クロス環境との接続

表 5-3 にプラットフォームである FPGA ボードの機能を示す。FPGA ボードと PC を接続するインタフェースのうち端末ソフトウェアとのシリアル通信には USB/JTAG/UART を使う。USB/JTAG/UART はビットストリームに変換した回路プログラムの FPGA へのダウンロードにも使用している。したがって、PC 上の端末ソフトウェアと FPGA ボードのシリアル通信に新たなインタフェース追加は必要ない。

FPGA ボード上の UART デバイスと FPGA デバイス間の UART 接続には送信と受信機能の HDL 記述 UART_Rx と UART_Tx を用意した。

表5-3 実験に使用したFPGAプラットフォームの機能

機能	仕様		用途
FPGA	Artix-7 (XC7A100T-1CSG324C)		FPGAデバイス
クロック供給	100MHzまたはMMCMによる		Mixed-Mode Clock Manager
メモリデバイス	128 Mb Quad-SPI Flash		
温度センサ	ADT7420		BTS
表示	8桁7セグメントLED LED×16 3色LED×2		データ表示 ステータス表示 動作モード表示
スイッチ	スライドSW×16 ボタンSW×5		パラメータ設定 動作指定
外部メモリ	MicroSD card		カードからの回路プログラム
外部通信	USB/JTAG/UARTポート		iMCM通信
外部拡張 モジュール	Pmod A Pmod B Pmod C Pmod D	UART_1 UART_2 温度センサ RTC	PU通信 PM通信 ATS接続 DS1302 (Maxim integrated)
外部電源	5.5 V		USBまたはDCジャックより

5.4.2 モニタ機能の動作確認

クロス環境と FPGA ボードを含む実験している様子の画像を図 5-8 に示す。画像手前中央付近に FPGA ボード、奥にクロス環境のホストである PC の表示画面に端末ソフトウェアがメモリ内容を一括表示 (dump) していることが分かる。図 5-9 に UART 接続と RTC 接続で機能拡張した FPGA ボードの画像を示す。

実装したモニタ機能の一つであるメモリ関連処理のうち、表示処理と変更処理時の端末ソフトウェアの動作画面を図 5-10 に示す。画面の各行の先頭文字 ">" は利用者に対する 1 行入力促進のプロンプト文字であり、行末までは利用者が入力したコマンドとオペランド類である。利用者の操作は、(1) メモリを初期化、(2) 内容を一括表示、(3) ワード単位で表示、(4) バイト単位で表示、(5) メモリ内容を変更、(6) アドレス逆順で一括表示、と進めた結果である。このように、利用者はインタラクティブにメモリの状態を知り、また変更できる。

同様に、すべてのコマンドに対してあらかじめ定めた機能の正常動作を、キー入力と画面表示により確認した。

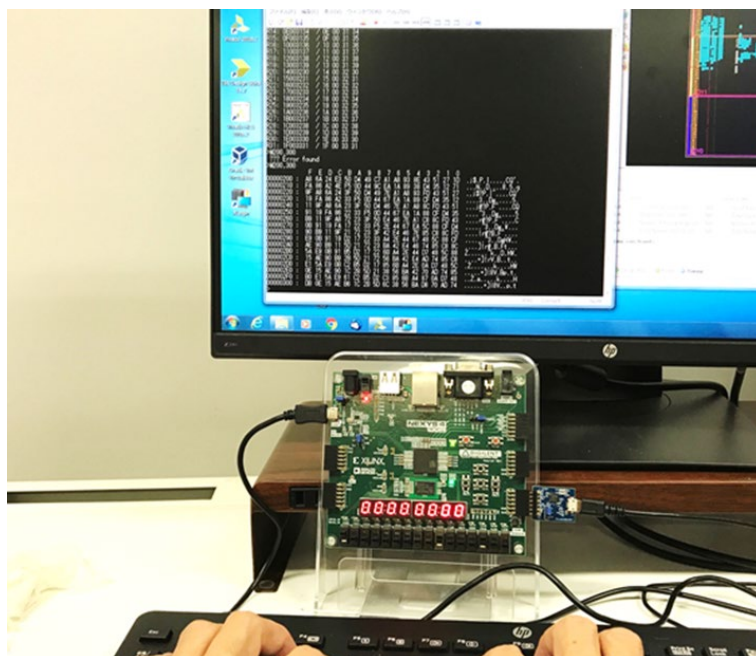


図 5-8 実験環境の画像

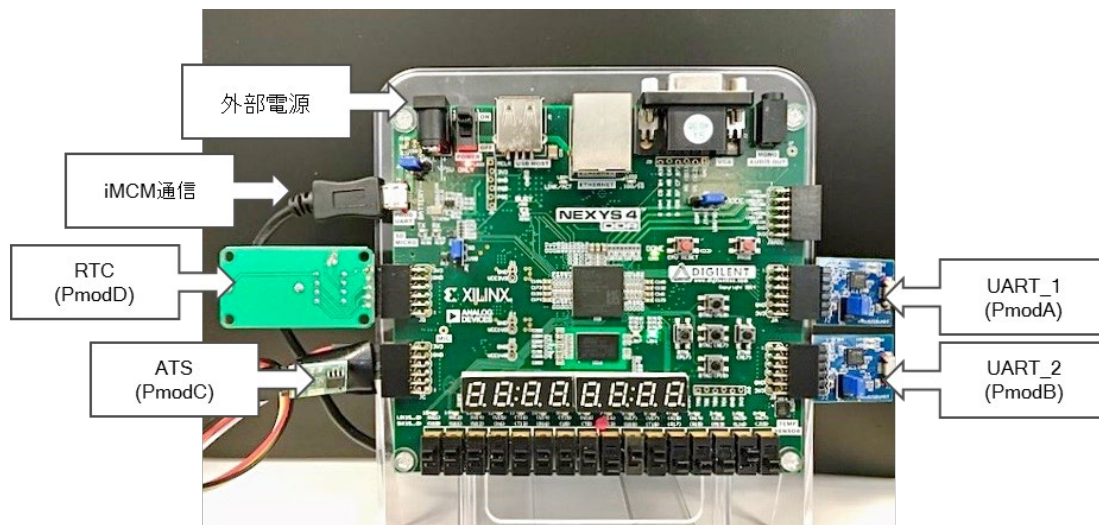


図 5-9 機能拡張した FPGA ボードの画像

ボード中央付近の FPGA はヒートシンク未装着であり、また自然空冷の状態

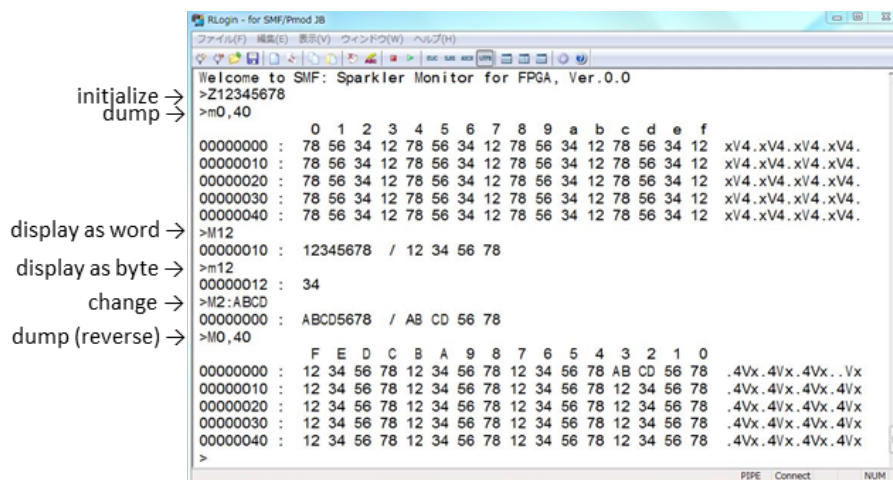


図 5-10 M (m) コマンドによるメモリ操作・表示の様子

5.4.3 実装結果

初期テストの実装結果

まず、初期テスト用に本来プロセッサ側に実装される 32bit 幅 RAM を用いた 0~16 KW (64 KB) のプロセッサメモリと 32bit×32 本のレジスタファイルをモニタ側に実装して RAM サイズの実装感度を確認した (図 5-11)。

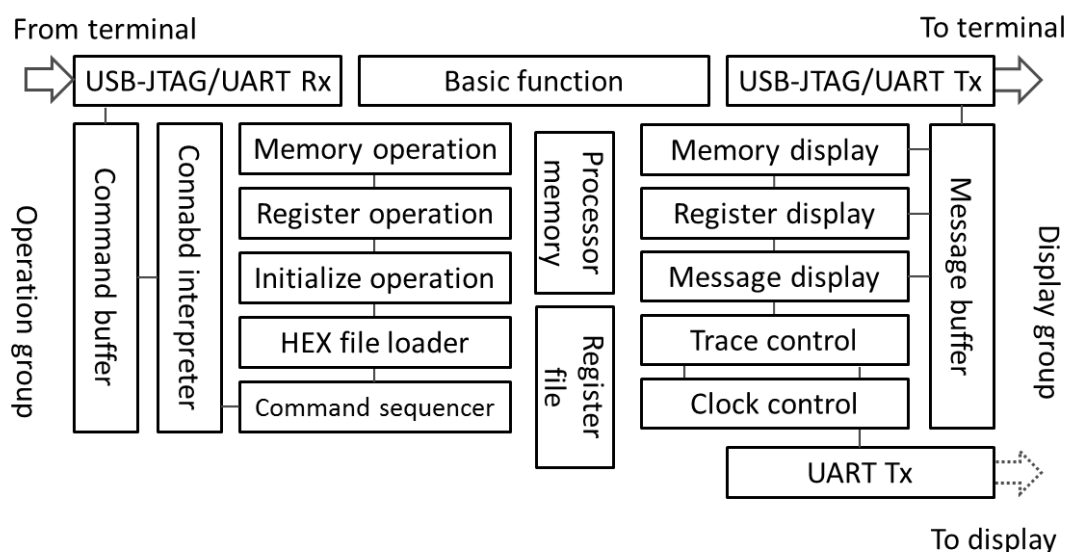


図 5-11 iMCM の単体動作構成

図 5-12 に HDL 実装時間 (論理合成に要した時間と配置配線に要した時間の合計) と FPGA 内の諸リソース占有率 (utilization) をグラフ化して示す. FPGA のリソースとなる構成要素には (1) LUT as logic, (2) LUT as RAM, (3) FF, (4) Block RAM, (5) DSP がある. 実験に用いた FPGA デバイスでは, 主要な構成要素のリソース量は SLICE 数への依存度が強い (図 5-13). そこで, 図 5-12 に示したように Utilization (LUT) で示す FPGA の代表的なリソースとして LUT 数で比較すると, iMCM 単独 (RAM サイズが 0) の場合では 3.3%, 統合を予定する 1 KW (4 KB) のプロセッサメモリを想定した場合では 5.6%と, iMCM にプロセッサを加えたとしても十分に実装できると考えられる.

一方 16 KW (64 KB) のプロセッサメモリを想定すると LUT 占有率は 30.1%と他回路の実装への圧迫が大きくなるばかりでなく, 配置配線を主として HDL 実装時間が 1 KW では 171 秒のところ, 16 KW では 392 秒と急激に長くなる実装時間となった.

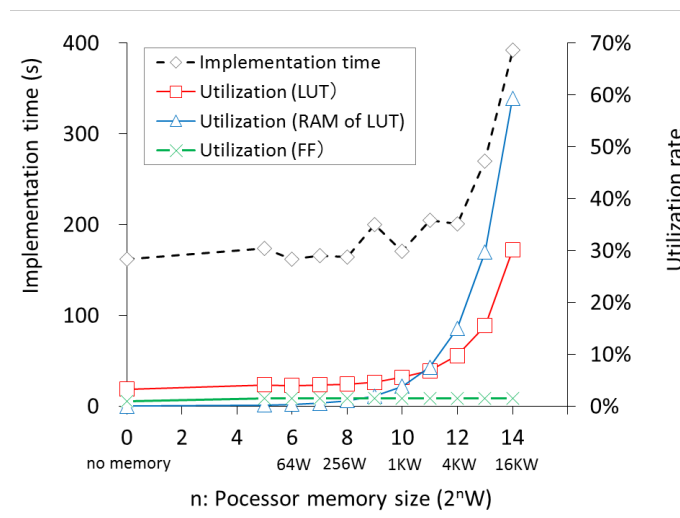


図 5-12 プロセッサメモリサイズに対する実装時間とリソース占有率の関係

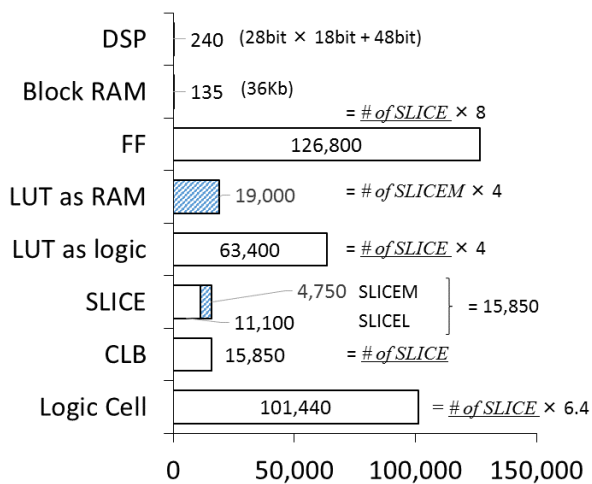


図 5-13 FPGA デバイス (XC7A100T) の構成要素数
DSP 数と Block RAM 数以外は SLICE 数から導かれる

コマンド機能の実装結果

L / M / R / Z コマンド機能のそれぞれを処理する SM の実装に必要な FPGA のリソース使用数を表 5-4 に示す。4 つのコマンド機能を含む全機能の実装に対して、入出力制御・文字列入出力・データ型変換などコマンド共通の基本機能実装時の LUT 占有率は 53.3% である。この値は使用した FPGA 全体の有効 LUT に対して 1.7% となる。5.1 で述べた今後接続を予定しているプロセッサの FPGA リソース量を加味しても、新たなコマンド機能の実装に備えるには十分小さいと言える。

表5-4 各FPGAリソース使用数

FPGA ブロック		使用数 [†] / 搭載数			
		LUT as logic	LUT as RAM	FF	Block RAM
		/ 63,400	/ 19,000	/ 126,800	/ 135
基本機能		1,107	0	574	1.0
メモリ関連	(M コマンド)	480	0	119	0.0
レジスタ関連	(R コマンド)	128	0	284	0.0
初期化関連	(Z コマンド)	65	0	105	0.0
ロード関連	(L コマンド)	298	0	75	0.5
フル機能	M+R+Z+L	2,078	0	1,157	1.5
全機能実装時の占有率 (%)		3.3	0	0.9	1.1

[†] 配置・配線後の数値

5.5 プロセッサ統合への検討

5.4 では iMCM の機能面の有効性と単体実装について述べた。ここでは、ターゲットプロセッサと iMCM の統合について述べる。

5.5.1 統合するプロセッサの要件

iMCM は組み込み機器向けの固有 ISA を実装する 16bit、及び 32bit プロセッサの中核部（コア）を組み込んだプロセッサへの接続を想定した。5.4 で述べた単体実装時のコアに接続するプロセッサメモリは、32bit 幅でクロック同期書込み・非同期読出しでタイミング設計した。iMCM は実際に統合するプロセッサのマイクロアーキテクチャの自由度を考慮し、32bit 幅または 16bit 幅のいずれか、及び非同期読出しまたは同期読出しのいずれかを選択できる HDL 記述を用意した。回路合成時に不要な HDL 記述をコメントアウトにより選択した。

5.5.2 統合の貢献対象と貢献時期

ここまで述べたように、iMCM はプロセッサの設計品質レベルと、ソフトウェア開発環境の整備レベルに依存せずに、機械語モニタの機能を提供できるばかりでなく、プロセッサメモリのタイミング設計に依存せず接続できる展開力を持つ。このような展開時には、それぞれのプロセッサの機能確認や低レベルプログラミングの優先順位と実装時期が異なる可能性がある。そこでそれぞれのプロセッサに対応して、iMCM が使用できる FPGA のリソースと必要なモニタ機能を最適化する必要がある。

FPGA のリソースに全機能の iMCM を集積する余裕があったとしても、使用しない機能を処理する回路の実装はエネルギー浪費の観点から好ましくない。このため、必要にして十分なモニタ機能を実現する回路のみの実装が推奨される。プロセッサの設計品質が高まるにつれて必要なモニタ機能を限定され、プロセッサが実際の運用状態になればモニタ機能は完全に削除できる。また逆に、応用システム向けプラットフォームの定期的な自己診断時や偶発故障の発生時などには、改めてプロセッサの機能・性能をトレードオフとして iMCM を再実装できる。

5.5.3 プロセッサ設計者へのニーズヒアリング

コアを組込むプロセッサ, 及びデータメモリを使用する専用プロセッサの設計経験者あるいは設計中の6名に, iMCM が提供するモニタ機能が「好ましく支持できるか?」をアンケート形式による予備ヒアリングを実施した. 回答者 A は大学院生であり, 他の回答者は大学生であった. 6名が設計したプロセッサの ISA は表 5-5 のとおりである. これらの ISA は, すべて FPGA にコアあるいはプロセッサとして FPGA に実装された. 支持の判定には, 表 5-6 に示すように, 機能説明への理解に対して共感できる標語の回答を得た. 明確な支持の回答により強い重みを付けた平均を求めた. iMCM が実装したモニタ機能とヒアリングの結果, 過半数の支持(「欲しい」と「あるといい」の2レベル)を得たモニタ機能を表 5-7 にて”✓”(checked)で示す. 回答をまとめた結果, FPGA へのプロセッサの実装に先立って, 相当なクロック数のシミュレーションができる性能の PC をホストとしたクロス環境が容易に入手できる近年においても, 機械語モニタへの期待があることを確認した.

表5-5 モニタ機能ヒアリング者リスト

ヒアリング者	FPGA実装	実装したISA (ISA内分類)
A	あり	MicroBlaze (AXIバス拡張)
B	あり	RISC-V (RV32GC, RV64GC)
C	あり	RISC-V (RV32Cベースにカスタマイズ)
D	あり	COMET II (命令拡張)
E	あり	MIPS (周辺機能接続)
F	あり	MIPS (EPIC [†] 化) [92]

[†] Explicitly Parallel Instruction Computing, VLIWを採用している

表5-6 モニタ機能アンケートの判定方式

平均点数が1.5以上“✓”判定	分類	当てはまる標語	点数
欲しい	過去の仮定	あったら使った	3
		あったら使ったかもしれない	1
あるといい	次の可能性	あるなら使う	3
		あるなら使うだろう	1

表 5-7 モニタ機能のヒアリング結果

カテゴリと機能	欲しい	あるといい	iMCM
実行制御			
実行開始番地の設定			
指定番地からの実行開始	✓		✓
指定番地での実行中断		✓	✓
実行中断からの復帰(実行継続)		✓	✓
1ステップ実行		✓	✓
指定した1命令の実行	✓		
トレース範囲となる番地の設定	✓		
メモリ関連			
メモリの初期化	✓		✓
メモリ内容の表示と変更	✓		✓
メモリ内容の一括表示(ダンプ)	✓		✓
メモリ内容の文字列表示		✓	✓
メモリ内容の一括転送			
メモリ内の指定データ検索	✓		
メモリ以外のコア資源関連			
レジスタの初期化	✓		✓
レジスタ内容の表示と変更	✓		✓
フラグ内容の表示と変更	✓		—
ターミナル, 周辺装置関連			
プログラム/データのロード		✓	✓
プログラム/データのストア			
ターミナル画面のハードコピー			—
ターミナルの変更			—
I/O へのアクセス(入力/出力)		✓	—
その他機能			
コマンドの連続実行	✓		✓
リアルタイムクロックとの連携/同期		✓	✓
トレース	✓		✓
簡易アセンブル	✓		
逆アセンブル	✓		

5.6 ターゲットプロセッサの概要

ターゲットプロセッサの命令セットアーキテクチャ

iMCM に統合させるターゲットプロセッサ (SPARKLER) は, 将来的に ISA を拡張して性能を向上させるためのプラットフォームである. 拡張のベースとなるのは, 最小限の ISA と単純なマイクロアーキテクチャである. 表 5-8 に示すように, SPARKLER では, オープン ISA RISC-V [47] の中で最小の 16bit 命令長を持つ圧縮命令セット拡張 (“C” 標準拡張: RV32C) [89] を採用した. SPARKLER と異なる RISC-V アーキテクチャの改造, 及び固有の実装について報告した [90], [91]. 表 5-8 の「命令」の行には, 実装されている 27 命令が記載されている. しかし, iMCM の実装では, `c.ebreak` 命令の機能がプログラムの実行を停止するように変更されている. デバッガはプログラム開発に有益なツールの一つである. しかし, プログラム開発の初期段階では, デバッガの利点を必要としない. そこで, iMCM に対応するように機能を変更し, 限られた命令コードの有効利用を試みた. マイクロアーキテクチャとしては, 命令処理時の消費電力を最小限に抑える実装を目指して単一サイクル方式を採用した. RV32C は基本命令セット RV32I の 32bit 長の命令の一部を 16bit に圧縮したものである.

表5-8 SPARKLERの命令セットアーキテクチャ

基本 ISA	RISC-V	
分類	RV32C	16bit 命令長, 32bit ワードデータ長
レジスタ		32 本 (32bit)
命令空間	命令 :	2^{30} ワード (32bit)
	データ :	2^{30} ワード (32bit)
命令†	演算 :	ADD, SUB, AND, OR, XOR, SLLI, SRLI, SRAI, ADDI, ADDI16SP, ADDI4SPN, ANDI
	ロード/ストア :	LW, LWSP, SW, SWSP
	分岐/ジャンプ :	J, JR, JAL, JALR, BEQZ, BNEZ, EBREAK
	転送 :	LI, LUI, MV, NOP

† ニモニク表現 <ADD> は <c.add> を置き換えた例

ターゲットプロセッサの設計

SPARKLER は RV32I の命令セットではなく、RV32C をネイティブの命令セットとし、命令デコードに関わる回路を最小限に抑制した。図 5-14 に SPARKLER のブロック図を示す。動作クロック (clk) は、プログラムカウンタ (PC) で指定された次のアドレスの同期にのみ使用されている。4 KB (32bit×1 KW=16bit×2 KHW) の命令メモリ (IM)、命令デコーダ (ID)、レジスタファイル (RF)、実行ユニット (EX)、4 KB (32bit×1 KW) のデータメモリ (DM)、ライトバック (WB) を介して、1クロック周期で命令実行が進む。

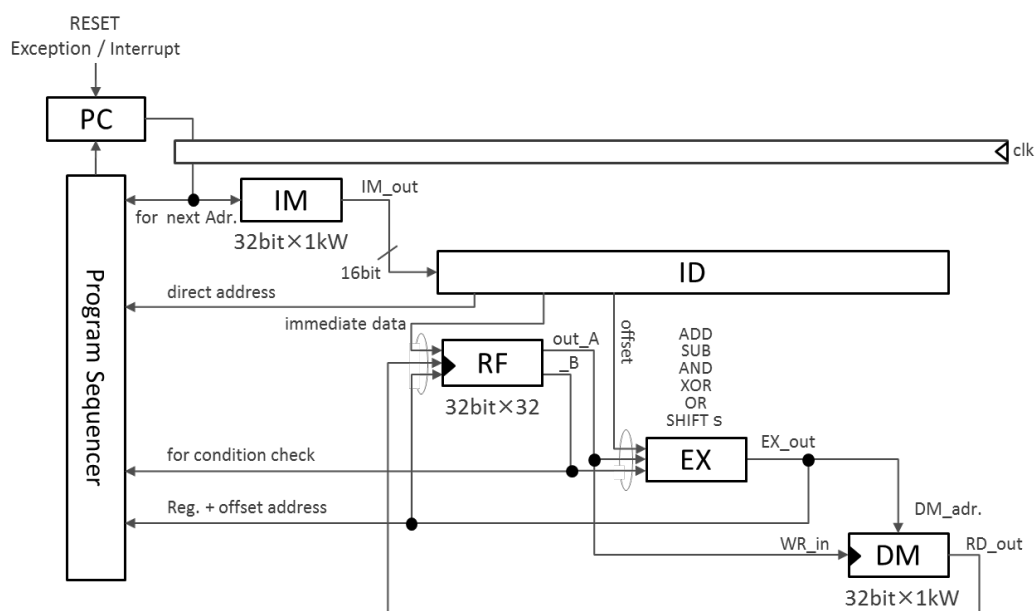


図 5-14 SPARKLER のブロック図

iMCM との統合

iMCM 統合のためのプロセッサコアへの追加・変更は、次のようにローカルなものであった。(1) プロセッサメモリとレジスタファイルの I/O ポート追加と ID への一部情報の追加,(2) 動作クロックの再構成,(3) 割込み要求/受付の再構成である。図 5-15 は、上記 (1) の具体的な実施例を示す図である。PC, IM, RF, DM に拡張ポート (ePORT) を追加し、ID にも拡張デコード情報 (eID) を追加した。

さらに、コアの追加や変更は、提供するモニタ機能の選択によって制限を受ける。コアを設計した後でも、一部の機能追加、及び信号の接続/切断により、上述した実施形態の再構成が可能である。eID には、トレース時に表示する命令のニモニック (mnemonic) 情報が含まれる。ニモニック情報には、命令名、一つ以上のオペランド名などが含まれる。ISA による情報セットの違いや命令数の違いは、eID の大きさに影響する。例えば、x86 アーキテクチャで表される Complex Instruction Set Computer : CISC のような複雑な ISA では、eID の大きさに応じてトレース機能の実装を検討する必要がある。

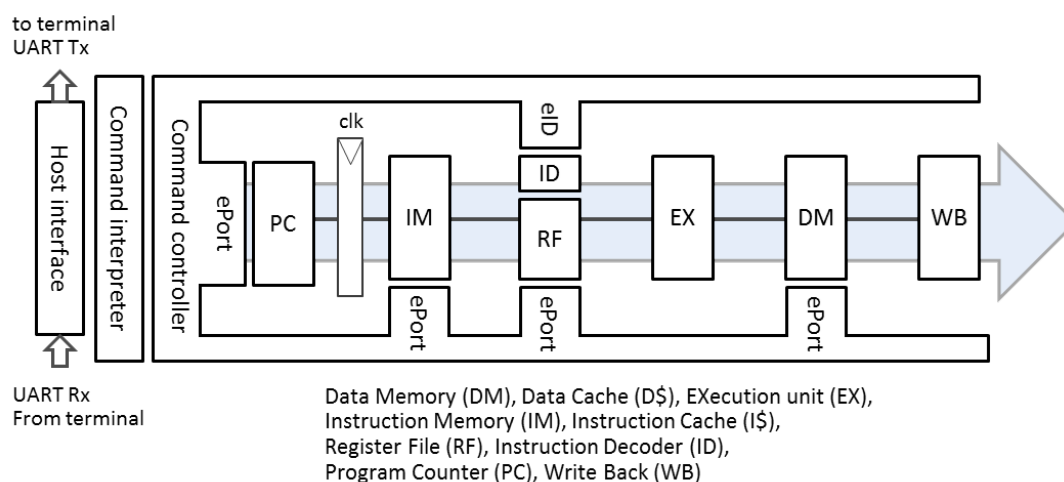


図 5-15 ターゲットプロセッサを単一クロック実装した構造

5.7 統合実装実験と評価

5.7.1 統合時特有の動作確認

図 5-16 は、SPARKLER 内部主情報をトレースした結果の表示画面である。画面の各行は、プログラムの 1 命令の実行に対応している。プログラムの実行状態として、クロック数、PC 値、命令コード (16 進数と 2 進数で)、ニモニック、主な命令デコード情報、操作すべきオペランドを含むプロセッサの状態が利用者に報告されている。一部のモニタ機能に共通する逐次処理や SM をシミュレータで検証した。

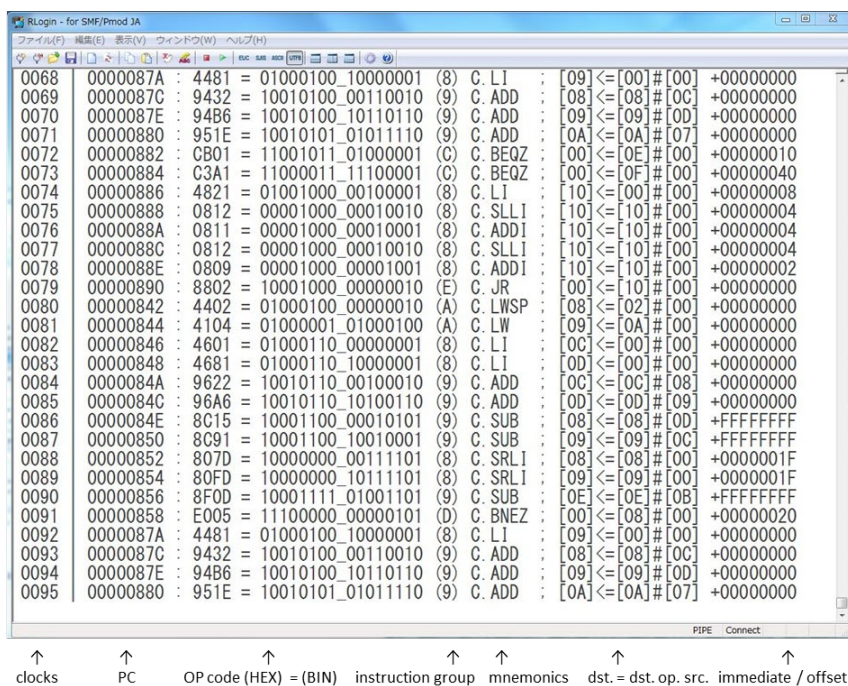


図 5-16 プログラムトレースの様子 (Fibonacci 数列計算)

5.7.2 提案機械語モニタのプロセッサ統合実装結果

表 5-9 に、プロセッサと iMCM の実装に必要な FPGA 内の LUT 使用数に基づく回路規模を示す。プロセッサの実装のために iMCM の実装に 41%の LUT を追加した。この追加は、選択するモニタ機能に依存する。この追加のうち 58%は、入出力制御、文字列入出力、データ型変換などのコマンドに共通する基本的な機能処理のためのものである。これが iMCM 実装のための最低限のリソース量となった。この値は、使用した FPGA 全体の有効 LUT の 2.2%であった。新しいコマンド機能を実装するために必要なリソースであった。IM と DM はハーバード・アーキテクチャ (Harvard architecture) で構成されている。それぞれ 4 KB を実装し、IM は 2,048 命令の容量を持っていた。IM の実装に必要な LUT 数は SPARKLER の占有 LUT の約 18%、全 LUT の 1.6%であった。実行するプログラムのサイズに応じて容量を拡大する余地がある。

表5-9 各FPGAリソース使用数に基づく回路規模

FPGA ブロック		使用 LUT 数
プロセッサ (SPARKLER)		5,979
iMCM (Integrated Machine Code Monitor)		2,465
基本機能		1,423
メモリ関連	(M コマンド)	480
レジスタ関連	(R コマンド)	128
初期化関連	(Z コマンド)	65
ロード関連	(L コマンド)	298
プログラムトレース (シングルステップを含む)		71
合計	63,400 (all)	8,444

5.8 考察

以上に示したように、iMCM は開発環境の準備状況に依存しないモニタ機能を提供でき、プロセッサとのインタフェースは疎結合である。将来的には、5.6 で述べたプロセッサの拡張版だけでなく、全く異なるプロセッサにも iMCM への接続が予想される。いずれの場合も、プロセッサの機能検証や低レベルプログラミングの優先順位や所要時間が異なる可能性がある。そのため、各プロセッサに対応して、(a) iMCM が使用できる FPGA のリソースと (b) 利用者が必要とするモニタ機能の両方を最適化する必要がある。

全 iMCM 機能を有する回路の実装は、エネルギー浪費の観点から好ましくない。必要なモニタ機能を実現する回路に限定する実装を推奨する。プロセッサ全体やコア部分の設計品質レベルが高くなればなるほど、必要なモニタ機能は限定的になる。プロセッサが実際の動作段階になれば、モニタ機能を完全に削除できる。

逆のケースとして、プロセッサ応用システムのプラットフォームの定期的な自己診断時や、ランダム障害が発生した場合に、iMCM の再実装が可能になる。このとき、モニタ機能は、機能とプロセッサの性能のトレードオフが必要な状態で iMCM に実装できる。

更に 5.2.3 で述べたように、プロセッサの全命令が正しく実装されていることが動作の前提である。この前提に対して、提案 iMCM は、プロセッサの一部の回路が故障している状態であっても、モニタ機能を利用者に提供できる。プロセッサ外部から、プロセッサメモリとレジスタファイルに直接アクセスするからである。

5.9 おわりに

固有の ISA を実装するプロセッサを FPGA に搭載する際に、プロセッサの設計品質レベルと、ソフトウェア開発環境の整備レベルに依存せず、プロセッサと同様に FPGA に回路として実装する機械語モニタを提案した。さらに提案モニタを FPGA 上に実装して評価し、モニタ機能群の単独有効性を確認した。

さらに、ISA に RISC-V をベースとするプロセッサに対応し、回路として FPGA 上に実装するための iMCM を提案した。iMCM に共通の SM をシミュレータでテストした。さらに、実際の FPGA を用いて、モニタ機能ごとに固有の SM をテストした。選択するモニタ機能に応じて、iMCM 実装時の回路規模は、プロセッサコア実装時に 24% から 41% の LUT を追加した。加えて、FPGA 評価ボード上の FPGA にプログラムされた回路の 100 MHz 動作を確認した。

第6章 結論

本章ではまず、前章までに述べた FPGA 搭載プロセッサの高温試験に関する一連の研究成果をまとめて総括し、最後に本研究の展望と課題について考察する。

6.1 本研究の成果

FPGA の温度環境を対象とする信頼性保証は、恒温槽を用いた信頼性試験、または開発環境を用いた設計保証で行われてきた。前者はコストの問題、後者はマージン最小化の困難さに問題があった。このため、トレードオフとしてコストと保証レベルを、ユーザが選択しなければならない問題を抱えていた。この問題を解決する一つの方式が、FPGA 自体を発熱させることである。この領域の従来研究には、高速・高性能 FPGA を用いた報告があるものの、低消費電力 FPGA では具体的な提案は見受けられなかった。

本研究では、低消費電力 FPGA を用いても高温試験に資する内部加熱器を提案した。実装した拡張駆動リングオシレータ加熱器は、装置の回路基板に搭載した状態で FPGA デバイス内部から加熱する。効率的な加熱を FPGA 構造特有の敷き詰められた配線ブロックと論理ブロックの構造に着目し、従来は駆動対象に考慮されなかったフィードバック信号を拡張駆動負荷として加熱に利用した。拡張駆動リングオシレータを用いた内部加熱器は、最大推奨動作温度 85°C を超え、絶対最大定格 125°C までチップを加熱する結果を示した。従来研究と実装した FPGA の種別が異なることを加味しても、提案した内部加熱器は FPGA リソースで同等、及び消費電力で効率的な加熱を示したといえる。

各章のまとめ

本論文の第1章では、本研究の背景として FPGA に関する信頼性保証の特徴と解決すべき課題を示し、本研究の目的について述べた。加えて、最近の FPGA の特徴を述べることにより、背景と目的を補足した。更に、本論文の構成と各章の概要についてまとめた。

第2章では、熱が FPGA にもたらす不利益な問題と、対極にある有益な利用について述べた。問題については、まず IC 全般にわたる共通点から述べた。利用については、FPGA 高温試験向けの温度環境を供する内部加熱器について述べた。

第3章では、内部加熱器を実装する FPGA の規模的な制約を背景として、搭載プロセッサの加熱能力について述べた。プログラムとデータの性質がおよぼす消費電力への影

響, 及び加熱への影響からチップ温度への関係を把握し, この関係のモデル化検討について述べた. また, レジスタ転送レベルから新たに定義する小型プロセッサを FPGA に実装し, 効率的にチップを加熱する発熱プログラムについて述べた. 更に, 4 種のプロセッサ動作速度と 5 種の発熱プログラムを組み合わせることにより, FPGA を加熱する条件について述べた.

第4章では, FPGA の高温試験を必要とする状況を背景として, リングオシレータを利用し, FPGA デバイス自体が柔軟に発熱する内部加熱器について述べた. 内部加熱器は最大定格が定める温度までのチップ加熱を目的とした. まず, 内部加熱器をプロセッサと統合する構成について述べた. 統合する構成を FPGA に実装し, 内部加熱器を構成する遅延素子の遅延段数と駆動段数, 加熱モジュール数の条件がチップ温度へ与える影響について述べた. 更に, プロセッサとともに実装した内部加熱器の特性について述べた. また, チップを推奨動作温度 85°C 以上に加熱する条件と, 最大定格が定める 125°C への加熱可能性について述べた. 加えて, 最短時間で目標チップ温度へ加熱する温度制御回路を FPGA に実装し, 評価結果に基づいてプロセッサの高温テストへの応用可能性について述べた.

第5章では, 高温環境下で回路の一部がマージン不足となる状況を背景として, 本来のモニタ機能を維持する実装手法について述べた. まず, ハードウェア記述言語で記述され, プロセッサとともに FPGA に実装する機械語モニタの実装手法について述べた. 提案する実装手法は, プロセッサの完全動作, 及びクロス環境の整備状況に依存しない. 更に, 機械語モニタの機能選択にあたって実施したプロセッサ設計経験者へのアンケートと, 回答結果について述べた. 加えて, 選択した機能とともに FPGA に実装したプロセッサについて述べた. このプロセッサは第3章および第4章において実装するプロセッサと同一である. 提案実装手法は, プロセッサが高温環境下でマージン不足となる一時的・部分的な故障状態にあっても, モニタ機能の提供を維持すると考えられる.

最後に本章では, 本研究の総括と今後の展開を述べた.

以上, 本研究は FPGA 搭載プロセッサの高温試験に関する一連の研究をまとめたものであり, これらの成果は高温試験の広範な応用に貢献できると考える.

6.2 今後の展望と課題

提案した内部加熱器は、LUT のみで実装する柔軟性に優れており、幅広い分野に応用可能である。今後の応用を拡大する課題は、温度制御の精度向上、及び個別環境に対応した最適な構成条件と制御条件の選択である。以下に、それぞれの課題に対する対応について述べる。

温度制御の精度向上

目標加熱温度に対する CUT 近傍のチップ温度差を、最短時間で最小にすることが精度向上に直結する。内部加熱器により加熱されたチップ温度は、チップ上の位置によって数°Cオーダの温度差がある。この温度差は、CUT の位置とチップ温度感知器の位置に対して、ユーザが内部加熱器の位置を明示的に指定して配置することで改善可能と考える。

個別環境に対応した最適条件の選択

加熱対象の FPGA、及び FPGA を搭載する回路基板の熱関連特性に応じて、内部加熱器の構成条件と温度制御回路の制御条件を調整する必要がある。FPGA および回路基板からの放熱に関する影響を調整すれば、相当の改善が期待できる。一方、設計情報からの条件調整では、複合的な放熱要因の精度よいモデル化に困難が予見される。最適条件の絞込みには、様々な実測結果を入力する深層学習の適用領域と考えられる。深層学習を補助する回路を再構成して実装すれば、FPGA が期待されている応用のひとつとして活用可能と考える。

本論文の最後に、さらなる発展として自律的に信頼性を維持する FPGA 実装について述べる。本研究では、高温試験の実施タイミングはユーザが決定する前提とした。本研究の発展には、高温試験に最適なタイミングを FPGA 応用製品自体が判定するとともに、FPGA 上の実装回路を対象とする高温試験の例が考えられる。最適なタイミングの決定には、加熱に必要な供給電力の余力検知がある。また、実装回路の高温試験に際しては、高性能プロセッサから低消費電力プロセッサへの再構成がある。

図 6-1 に具体的な構成案を示す。図 6-1 (a) は通常の状態の構成を示す。生存タイマ (survival timer) あるいはエラーカウンタ (error counter) が、プロセッサの

動作に不安定さがないかなど信頼性の程度を検知する。信頼性低下の気配を感知したタイミングで、図 6-1 (b) に示す試験状態に遷移する。試験状態では内部加熱器を構成し、プロセッサを高温で試験する。動作マージンの減少を検知して、性能重視から低消費電力重視のプロセッサを再構成する。課題であるタイミング決定と、再構成手法の解決により、装置レベルの信頼性を高くすることにまで貢献するだろう。

以上述べたように、FPGA の特長を活かした展開が期待できる。

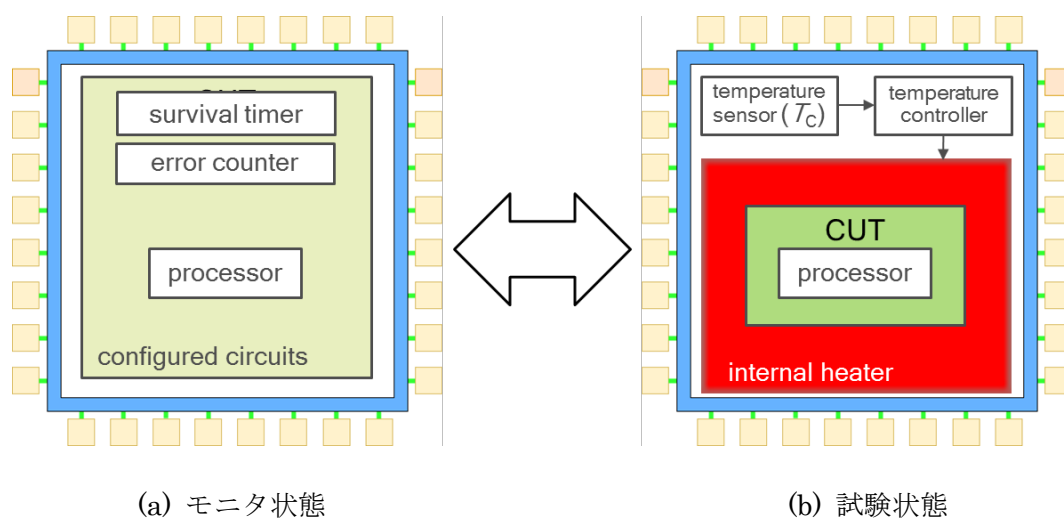


図 6-1 自律的に高温試験を行う FPGA

謝 辞

本研究の全過程を通じて、研究の方向付けから詳細に至るまで、辛抱強く懇切なるご指導とご鞭撻を賜りました、東京電機大学先端科学技術研究科電気電子システム工学専攻 金杉昭徳教授に心から感謝の意を表します。

本論文をまとめるにあたり、貴重なご助言を頂きました東京電機大学小松聡教授、田所貴志教授、和田成夫教授、佐藤修一准教授に厚く御礼申し上げます。

また、二度目の学生生活を集積情報システム研究室メンバの一人として、筆者が新鮮に過ごすことができた修士学生・学部学生のみなさんに感謝いたします。機械語モニタのアンケートでは、塚原彰彦氏、牧野俊也氏、三谷勇介氏に協力していただきました。

故大井尚一先生（東京電機大学 名誉教授）には、一度目の学生生活にてご指導いただきました。残念ながら 2016 年に逝去され、本論文に目を通していただくことは叶いませんでした。ここに、改めてご冥福をお祈りするとともに、深く感謝申し上げます。

あわせて、NEC、およびルネサスエレクトロニクスでの社会人時代に培えた多くの先輩・同僚・後輩にも本研究を側面から支えられました。1981 年に新社会人として NEC に入社して以来、ご助言を頂き続けている佐々木元氏（元 NEC 代表取締役会長）には感謝の念に堪えません。

そして、定年退職後に再度の学生生活を支えてくれた妻 亮子に感謝します。二度目の学生生活も健康に過ごせたことは筆者一人では叶わなかったことであり、理解と支えがなければ本論文を書き上げられませんでした。本当にありがとう。

以上、これらの方々のご理解とご協力なくして本論文は完成し得ませんでした。博士論文を書き上げられたことに対して、お名前を挙げられなかった方々も含めて、これまでお世話になった全ての皆さまに改めて心より謝意を表します。

参考文献

- [1] 石井翔平, 小林和淑, “商用 FPGA のばらつきと BTI による経年劣化,” 信学技報, vol. 112, no. 320, VLD2012-72, pp. 75-80, 2012.
- [2] 山口聖貴, Y. Yang, 坂本良太, 室山真徳, 石原亨, 安浦寛人, “90nmCMOS 回路における遅延・電力ばらつきのゲート段数およびゲート幅依存性に関する考察,” 信学技報, vol. 106, no. 550, ICD2006-209, pp. 45-50, 2007.
- [3] 白井康之, 丸山満徳, “カーエレクトロニクスの熱設計,” Hybrids, ハイブリッドマイクロエレクトロニクス協会, vol. 7, issue 4, pp. 39-43, 1991.
- [4] 田中裕史, “自動運転に求められるカーエレクトロニクスの信頼性,” ICT 戦略特別セミナー, 2017.
- [5] JEITA, IC ガイドブック よくわかる! 半導体 (2012 年版), 東京: 産業タイムズ社, 2012.
- [6] 矢野陽一, “32 ビット・マイコン「V60」開発物語,” 半導体産業人協会 会報, no. 7, pp. 17-20, 2012.
- [7] 川上雄一, “デジタル信号処理用プロセッサの開発,” 半導体産業人協会 会報, no. 7, pp. 13-16, 2012.
- [8] 桑田孝明, シリコンロジックプロセスの開発ものがたり, 東京: 日本半導体歴史館, 2015.
- [9] 阪本利司, 帰山隼一, 水野正之, 寺部一弥, 長谷川剛, 青野正和, “LSI 回路の再構成を可能とするナノブリッジ,” NEC 技報, vol. 60 (2007), no. 1, pp. 73-76, 2007.
- [10] 白旭, 阪本利司, 宮村信, 多田宗弘, “微細 CMOS と混載可能な Cu 原子スイッチ,” 応用物理, vol. 88 (2019), no. 2, pp. 115-119, 2019.
- [11] 富士キメラ総研, 先端/注目半導体関連市場の現状と将来展望 2017, pp. 87-96, 2016.
- [12] 富士キメラ総研, 自律制御・ロボット関連市場の現状と将来展望 2018, pp. 3-10, 2017.
- [13] 富士キメラ総研, 車載電装デバイス&コンポーネンツ総調査 2018 下巻 (ECU 関連デバイス編), pp. 165-168, 2018.
- [14] N. DOCOMO, White paper 5G Evolution and 6G, Tokyo, 2020.
- [15] N. DOCOMO, ホワイトペーパー: 5G の高度化と 6G, 2.0 版, 東京, 2020.

- [16] Xilinx, 7 series FPGAs Configurable Logic Block User Guide, UG474 (v1.8), 2016.
- [17] 尼崎太樹, 天野英晴 (編), FPGA の原理と構成, 東京: オーム社, 2016.
- [18] 石井翔平, 小林和淑, “90nm プロセス商用 FPGA にマッピングしたリングオシレータの発振周波数の劣化評価,” 信学技報, vol. 112, no. 324, VLD2011-55, pp. 19-24, 2011.
- [19] 松本智, “低温 CMOS コンピュータ,” 低温工学, vol. 25, no. 3, pp. 142-149, 1990.
- [20] 金子博昭, 金杉昭徳, “消費エネルギーを分配した領域毎に立上げ可能なプロセッサの検討,” 信学技報, RECONF2016-68, 2017.
- [21] ルネサスエレクトロニクス, 信頼性ハンドブック, R51ZZ0001JJ0250 (REV.2.50), 2017.
- [22] EIA/JEDEC, EIA/JESD22-A-104-F: Temperature cycling, Arlington, VA, USA, 2020.
- [23] D. Edwards, H. Nguyen, Semiconductor and IC Package Thermal Metrics, Application Report, SPRA953C, Texas Instruments Inc., 2016.
- [24] 国立天文台, “熱伝導率,” 理科年表 平成 30 年 机上版, pp. 425-427, 東京: 丸善出版, 2017.
- [25] EIA/JEDEC, EIA/JESD51-3: Thermal test board design with a low effective thermal conductivity for leaded surface mount packages, Arlington, VA, USA, 1996.
- [26] EIA/JEDEC, EIA/JESD51-5: Extension of Thermal Test Board Standards for Packages with Direct Thermal Attachment Mechanisms, Arlington, VA, USA, 1999.
- [27] EIA/JEDEC, EIA/JESD51-7: Thermal test board design with a high effective thermal conductivity for leaded surface mount packages, Arlington, VA, USA, 1999.
- [28] ルネサスエレクトロニクス, 半導体故障率, C11178JJCVOIF00 (第 12 版), 2010.
- [29] Xilinx, Vivado Design Suite User Guide: Programming and Debugging, UG908 (v2019.2), 2019.
- [30] Xilinx, XPE Xilinx Power Estimator User Guide, UG440 (v2017.1), 2017.
- [31] Xilinx, 7 Series FPGAs and Zynq-7000 SoC XADC Dual 12-Bit 1 MSPS Analog-to-Digital Convert User Guide, UG480 (v1.10.1), 2018.
- [32] 野口宏一朗, 野瀬浩一, 水野正之, “高信頼な LSI を実現するための微小遅延欠陥検出技術,” 信学技報, vol. 108, no. 140, ICD2008-42, pp. 23-28, 2008.

- [33] 野口宏一朗, 野瀬浩一, 尾野年信, 水野正之, “高機能 FF による微小遅延故障検出技術の小面積化手法,” 信学技報, vol. 109, no. 95, DC2009-16, pp. 31-34, 2009.
- [34] 澤田颯斗, 小松聡, “タイミングエラー見逃しを緩和するカナリアフリップフロップの提案と評価,” DA シンポジウム 2018 論文集, 石川県加賀市, pp. 130-135, 2018.
- [35] Y. Kameda, M. Mizuno, “Tunable duplex LSIs achieved by multiple phase-modulated clocks capable of predicting delay-increase and -decrease faults,” Symposium on VLSI Circuits 2009, Kyoto, Japan, pp. 114-115, 2009.
- [36] K. Noguchi, K. Nose, T. Ono, M. Mizuno, “A small-delay defect detection technique for dependable LSIs,” Symposium on VLSI Circuits 2008, Honolulu, HI, USA, pp. 64-65, 2008.
- [37] 金子博昭, 金杉昭徳, “FPGA 搭載プロセッサのダイ温度からの消費電力推定,” 信学技報, ICD2018-22, pp. 53-58, 2018.
- [38] H. Kaneko, A. Kanasugi, “Power consumption estimation by a die temperature for processors implemented on an FPGA,” AROB 2019, Beppu, Oita, GS13-2, pp. 343-346, 2018.
- [39] 伊藤謹司, 国峰尚樹, トラブルをさけるための電子機器の熱対策設計(第2版), 東京: 日本工業新聞社, 2016.
- [40] S. Bhagavatula, B. Jung, Y.H. Lu, “Real-time power sensors for intelligent power management and beyond,” IEEE Design and Test, vol. 31, issue 4, pp. 27-35, 2014.
- [41] W. Wu, L. Jin, J. Yang, P. Liu, S.X.D. Tan, “Efficient power modeling and software thermal sensing for runtime temperature monitoring,” ACM Transactions on Design Automation of Electronic Systems, vol. 12, issue 3, no. 25, pp. 1-29, 2007.
- [42] K. Memmi, G. Devogeleer, P. Jouvelot, F. Coelho, “Modeling the temperature bias of power consumption for nanometer-scale CPUs in application processors,” SAMOS 2014, Agios Konstantinos, Greece, pp. 172-180, 2014.
- [43] A. Porterfield, R. Fowler, S. Bhalachandra, W. Wang, “OpenMP and MPI application energy measurement variation,” E2SC 2013, Denver, CO, USA, no. 7, 2013.
- [44] C.H. Lim, W.R. Daasch, G. Cai, “A thermal-aware superscalar microprocessor,” IQSED 2002, San Jose, CA, USA, pp. 517-522, 2002.
- [45] Xilinx, Artix-7 FPGAs Data Sheet: DC and AC Characteristics, DS181 (v1.25), 2018.
- [46] Xilinx, 7 Series FPGAs Data Sheet: Overview, DS180 (v2.6), 2018.
- [47] K. Asanovic, D.A. Patterson, Instruction sets should be free: The case for RISC-V, Technical Report no.UCB/EECS-2014-146, University of California

- at Berkeley, 2014.
- [48] M. Gautschi, P.D. Schiavone, A. Traber, I. Loi, A. Pullini, D. Rossi, E. Flamand, F.K. Gurkaynak, L. Benini, “A near-threshold RISC-V core with DSP extensions for scalable IoT Endpoint Devices,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 10, pp. 2700-2713, 2017.
 - [49] P.V.S.R. Bharadwaja, K. Ravi Teja, M. Naresh Babu, K. Neelima, “Advanced low power RISC processor design using MIPS instruction set,” *ICECS 2015, Coimbatore, India*, pp. 1252-1258, 2015.
 - [50] 金子博昭, 金杉昭徳, “FPGA チップ温度制御用リングオシレータ加熱器の実装と評価,” *電子情報通信学会論文誌 C*, vol. J103-C, no. 11, pp. 473-482, 2020.
 - [51] M.K. Mandal, B.C. Sarkar, “Ring Oscillators: Characteristics and applications,” *Indian Journal of Pure and Physics*, 第 卷 vol.48, pp. 136-145, 2010.
 - [52] 三宅庸資, 佐藤康夫, 梶原誠司, “FPGA のリングオシレータを利用した温度モニタ,” *信学技報*, vol. 114, no. 384, DC2014-68, pp. 7-12, 2014.
 - [53] 三宅庸資, 佐藤康夫, 梶原誠司, “リングオシレータを用いた FPGA 上の完全デジタル温度モニタ,” *DA シンポジウム 2015 論文集*, 石川県加賀市, pp. 205-210, 2015.
 - [54] N. Zhu, Y. Zhou, H. Liu, “Counteracting leakage power analysis attack using random ring oscillators,” *2013 International Conference on Sensor Network Security Technology and Privacy Communication System (SNS & PCS)*, Harbin, China, pp. 12-20, 2013.
 - [55] P. Weber, M. Zagrabski, P. Musz, K. Kepa, M. Nikodem, B. Wojciechowski, “Configurable heat generators for FPGAs,” *2014 International Workshop on Thermal Investigations of ICs and Systems (THERMINIC)*, London, UK, pp. 1-4, 2014.
 - [56] A. Amouri, J. Hepp, M. Tahoori, “Built-in self-heating thermal testing of FPGAs,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, issue 9, pp. 1546-1556, 2016.
 - [57] 日本試験工業会, *JTM K07 (温度試験槽—性能試験方法及び性能表示方法) 規格*, 東京, 2007.
 - [58] 日本試験工業会, *JTM K09 (温湿度試験槽—性能試験方法及び性能表示方法) 規格*, 東京, 2009.
 - [59] 井上藤男, “恒温槽の温度均一化技術とその応用,” *島津評論*, vol.57, pp. 103-108, 2000.
 - [60] 石田俊己, “新 JTM 規格— 温度試験槽の性能試験方法及び性能表示方法,” *エスベック技術情報*, no. 49, pp. 12-20, 2007.
 - [61] 山品正勝, “配線技術と LSI 設計,” *まてりあ (日本金属学会会報)*, vol. 26, no.

- 6, pp. 555-560, 1997.
- [62] 寺島真一郎, “トランジスタ活用のための基礎知識,” 東芝トランジスタ回路集, pp.11-30, 東京: 誠文堂新光社, 1969.
- [63] ルネサスエレクトロニクス, RL78/F13,F14 ユーザーズマニュアル ハードウェア編, R01UH0368JJ0210 (REV.2.10), p. 1748, 2015.
- [64] A. Kanda, T. Kurafuji, K. Takeda, T. Ogawa, Y. Taito, K. Yoshihara, M. Nakano, T. Ito, H. Kondo, T. Kono, “A 24MB Embedded Flash System Based on 28nm SG-MONOS Featuring 240MHz Read Operations and Robust Over-The-Air Software Update for Automotive,” Symposium on VLSI Circuits 2019, yoto, Japan, pp. C200-C201, 2019.
- [65] I. Kuon, R. Tessier, J. Rose, “FPGA Architecture: Survey and Challenges, Foundation and Trend in Electric Design Automation,” Now Publishers Inc., vol. 2, no. 3, pp. 135-253, 2008.
- [66] 金子博昭, 金杉昭徳, “FPGA 搭載プロセッサ一体型機械語モニタ,” 信学技報, RECONF2017-61, pp. 65-70, 2017.
- [67] 村上晃司, 黒木幸令, “チップ内部の温度を測定可能な半導体センサの設計試作,” 平成 19 年度電気関係学会九州支部連合大会, 沖縄県那覇市, 08-1A-03, p.77, 2007.
- [68] 一色尚次, 北山直方, 伝熱工学 (新装第 2 版), 東京: 森北出版, 2018.
- [69] Xilinx, White Paper: Spartan-6 and Virtex-6 Devices, Power Consumption at 40 and 45nm, WP298(v1.0), 2009.
- [70] Xilinx, White Paper: Spartan-7 FPGAs, WP488(v1.0), 2017.
- [71] H. Kaneko, A. Kanasugi, “An integrated machine code monitor for a RISC-V processor on an FPGA,” AROB 2019, Beppu, Oita, GS13-1, pp. 339-342, 2019.
- [72] H. Kaneko, A. Kanasugi, “An integrated machine code monitor for a RISC-V processor on an FPGA,” Journal: Artificial Life and Robotics, vol. 25, no. 3, pp. 427-433, 2020.
- [73] T. Lange, B. Fiethe, H. Michel, H. Michalik, K. Albert, J. Hirzberger, “On-board processing using reconfigurable hardware on the solar orbiter PHI instrument,” 2017 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), Pasadena, CA, USA, pp. 186-191, 2017.
- [74] S.M.A. Raj, M.H. Supriya, “Underwater Image Enhancement using Single Scale Retinex on a Reconfigurable Hardware,” 2015 International Symposium on Ocean Electronics (SYMPOL), pp.1-5, Kochi, India, 2015.
- [75] D. Koeplinger, R. Prabhakar, Y. Zhang, C. Delimitrou, C. Kozyrakis, K. Olukotun, “Automatic Generation of Efficient Accelerators for Reconfigurable Hardware,” 2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture, Seoul, South Korea, pp. 115-127, 2016.

- [76] A. Kamaleldin, S. Hosny, K. Mohamed, M. Gamal, A. Hussien, E. Elnader, A. Shalash, A.M. Obeid, Y. Ismail , Y. Ismail, “A Reconfigurable Hardware Platform Implementation for Software Defined Radio using Dynamic Partial Reconfiguration on Xilinx Zynq FPGA,” 2017 RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing (NCSP'17), Guam, USA, pp. 209-211, 2017.
- [77] Y. Matsui, A. Tsukahara , A. Kanasugi, “An FPGA Implementation of Template Matching Processor,” 2017 RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing (NCSP'17), Guam, USA, pp. 209–211, 2017.
- [78] Y. Minagi , A. Kanasugi, “A Processor with Dynamically Reconfigurable Circuit for Floating-Point Arithmetic,” World Academy of Science, Engineering and Technology, International Journal of Electronics and Communication Engineering, vol. 4, no. 8, pp. 1269-1273, 2010.
- [79] Xilinx, MicroBlaze Processor Reference Guide, UG984 (v20191), 2019.
- [80] Altera, Arria V Device Overview, AV-501001, 2020.
- [81] Xilinx, Zynq-7000 All Programmable SoC Data Sheet, DS190 (v1.11), 2018.
- [82] Intel, Intel FPGA Monitor Program Tutorial for Nios II, 2017.
- [83] 松崎勉, “各種開発用システムの機能と新動向,” 日経エレクトロニクス, no. 121, pp. 118-135, 東京: 日経マグロウヒル, 1975.
- [84] 日本電気, “PDA-800,” マイクロコンピュータシリーズ総合ユーザーズ・ガイド, IEM-759A, pp. 245-248, 東京, 2007.
- [85] C64 Wiki, “Machine Code Monitor,” [オンライン]. Available: https://www.c64wiki.com/wiki/Machine_Code_Monitor#Features. [アクセス日: 1 12 2017].
- [86] Wikipedia, “ Machine code monitor,” [オンライン]. Available: https://en.wikipedia.org/wiki/Machine_code_monitor. [アクセス日: 1 12 2017].
- [87] Intel, Hexadecimal Object File Format Specification, Revision A, 1988.
- [88] Digilent, FPGA Board Reference Manual, Nexys4 DDR rev. C; Revised April 11, 2016.
- [89] D.A Patteson , A. Waterman, RISC-V Reader: An Open Architecture Atlas, Strawberry Canyon, 2017.
- [90] 三谷勇介, 金子博昭 , 金杉昭徳, “RISC-V の 16 ビット命令をベースとした組込み用途向け VLIW プロセッサの設計,” 信学技報, ICD2018-77, 2018.
- [91] 牧野俊也, 金子博昭 , 金杉昭徳, “RISC-V に基づく低消費電力プロセッサの開発,” 信学技報, ICD2018-78, 2018.

- [92] T. Hayashi , A. Kanasugi, “A design of EPIC type processor based on MIPS architecture,” *Journal: Artificial Life and Robotics*, vol. 25, no. 1, pp. 59-63, 2020.

本研究に関する発表文献

本研究に関する学術雑誌論文

- [1] 金子博昭, 金杉昭徳, “FPGA チップ温度制御用リングオシレータ加熱器の実装と評価,” 電子情報通信学会論文誌 C, vol.J103-C, no.11, pp.473-482, Nov. 2020.
- [2] H. Kaneko, A. Kanasugi, “An integrated machine code monitor for a RISC-V processor on an FPGA,” Journal: Artificial Life and Robotics, vol.25, no.3, pp.427-433, Aug. 2020.

本研究に関する国際会議論文

- [1] H. Kaneko, A. Kanasugi, “Power consumption estimation by a die temperature for processors implemented on an FPGA,” AROB 2019, Beppu, Oita, GS13-2, Jan. 23, 2019.
- [2] H. Kaneko, A. Kanasugi, “An integrated machine code monitor for a RISC-V processor on an FPGA,” AROB 2019, Beppu, Oita, GS13-1, Jan. 23, 2019.

本研究に関する学会研究会資料等

- [1] 金子博昭, 金杉昭徳, “消費エネルギーを分配した領域毎に立上げ可能なプロセッサの検討,” IEICE, RECONF2016-68, Jan. 24, 2017.
- [2] 金子博昭, 金杉昭徳, “FPGA 搭載プロセッサ一体型機械語モニタ,” IEICE, RECONF2017-61, Jan. 18, 2018.
- [3] 金子博昭, 金杉昭徳, “FPGA 搭載プロセッサのダイ温度からの消費電力推定,” IEICE, ICD2018-22, Aug. 7, 2018.
- [4] 三谷勇介, 金子博昭, 金杉昭徳, “RISC-V の 16 ビット命令をベースとした組込み用途向け VLIW プロセッサの設計,” IEICE, ICD2018-77, Dec. 13, 2018.
- [5] 牧野俊也, 金子博昭, 金杉昭徳, “RISC-V に基づく低消費電力プロセッサの開発,” IEICE, ICD2018-78, Dec. 13, 2018.