

オープンデザイン支援の最後のピース: Google の取り組みの変遷

招待論文

Tim Ansell

tansell@google.com

Google

カリフォルニア州マウンテンビュー

Mehdi Saligane

mehdi@umich.edu

ミシガン大学

ミシガン州アナーバー

要旨

オープンソースの電子設計自動化 (EDA) とハードウェア設計コミュニティを推進するイニシアチブにおいて、Google は、学界、スタートアップ企業、ファウンドリの研究者が関与するグローバルな共同の取り組みを牽引している。オープンソースシリコンの実現を最終目標に、発展中の複数のプロジェクトを支援し、EDA ツールの障壁、そして究極的にはハードウェア設計の障壁を開拓するというオープンソースの新しい波を導いている。ハードウェア民主化を推進するこの活動は、最先端の技術ノードで複雑かつ安全なシステムオンチップ (SoC) を迅速設計する基盤となる、シリコン実証済みのアナログおよびデジタル IP ブロックのオープンソースプラットフォームを開発およびリリースすることを目指すものもある。本稿は、オープンデザイン実現のために、欠けている最後のピースを形成するためのさまざまな取り組みとして、EDA ツールの OpenROAD、OpenRAM などのデジタルライブラリ、スタンダードセル、SoC 用のアナログ / ミックスシグナル (AMS) ビルディング・ブロック (BAG、FASoC など) について解説する。

CCS の概念

- ハードウェア → EDA 用ソフトウェアツール

キーワード

OpenROAD、オープンソース、VLSI、EDA、シリコンコンパイラ、Yosys、SystemVerilog、Verible、UHDM、Surelog

ACM 参考文献書式:

Tim Ansell and Mehdi Saligane. 2020. The Missing Pieces of Open Design Enablement: A Recent History of Google Efforts: Invited Paper. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD '20), November 2–5, 2020, Virtual Event, USA*. ACM, New York, NY, USA, 8p. <https://doi.org/10.1145/3400302.3415736>

1 序論

ネイサン・ミアボルドが提唱した「ソフトウェアの 4 つの法則」[1] は、ソフトウェアは利用可能なあらゆる演算能力を消費し尽くすまで拡張するとした。この消費によって、コンピュー

本稿は、クリエイティブコモンズの表示 4.0 国際版 (CC-BY 4.0) ライセンスにより公開されています。著者は、適切な帰属表示をしたうえで、この著作物を個人および企業のウェブサイト上で広める権利を留保します。この著作物に含まれる第三者のコンポーネントの著作権は守らなければなりません。その他の使用については、所有者 / 著者にご連絡ください。

ICCAD '20, November 2–5, 2020, Virtual Event, USA

© 2020 著作権は所有者 / 著者が保有します。

クリエイティブコモンズ CC-BY 4.0 ライセンスにより公開されています。

ACM ISBN 978-1-6654-2324-3

<https://doi.org/10.1145/3400302.3415736>

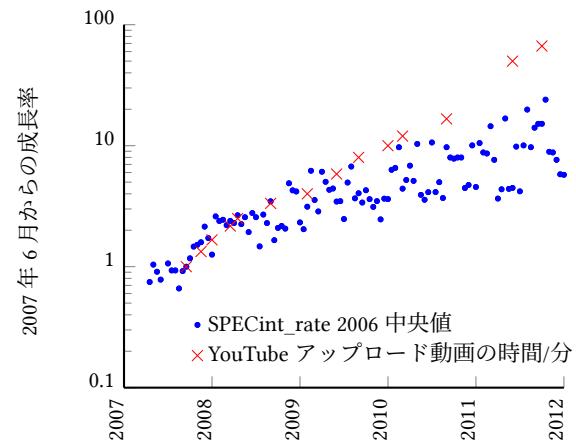


図 1: YouTube のアップロード動画の伸びと CPU パフォーマンスの比較

タ業界は、より少ないソフトウェアエンジニアリングのリソースで、従来以上に複雑で魅力ある製品の創出を可能にしてきた [2, 3]。これまでムーアの法則の安定した実現によって、この拡張が支えられてきたが、近年はこの法則が鈍化してきている。その一方で、演算能力に対する需要は増加するばかりだ。このような需要の継続的増加は、Google を始めとする多くのテクノロジー企業で見られる。図 1 は、高圧縮の YouTube 動画のコード変換に費やされたサイクル数をコンピューティングパフォーマンスと比較して追跡したものである [4]。

ムーアの法則に関する近年の鈍化が、準備不足の企業に与えた影響については、[5] に記載する。このため、先進的な企業は、ハードウェアアクセラレータ ASIC の開発など、この需要の増加に取り組むための新たなアプローチの調査にすでに着手している。なかでも Google は、Tensor Processing Unit (TPU) のようなハードウェアアクセラレータの開発で多大な成功を収めている [6]。このように Google の ASIC 開発への参画が進む中、図 2 に示すように、理論上可能なトランジスタ密度と、EDA ツールにより実現している実際のトランジスタ密度の差がますます重要になってきている [7]。

2 共同研究エコシステムの要件

Google の研究理念の原点は次のようなものである。

何よりもまず、知識の共有により、すべての人の進歩が加速する。[2]

この原則は、ハードウェア開発分野での Google の取り組みの基本となるものである。他の研究者の成果を再現し、その上に構築できることは、信頼できる知識共有のエコシステムに求められる根本的な要素である [8]。コンピュータやエレクトロニクス産業の発展に必要な計算集約型の研究分野では、研究結果の再

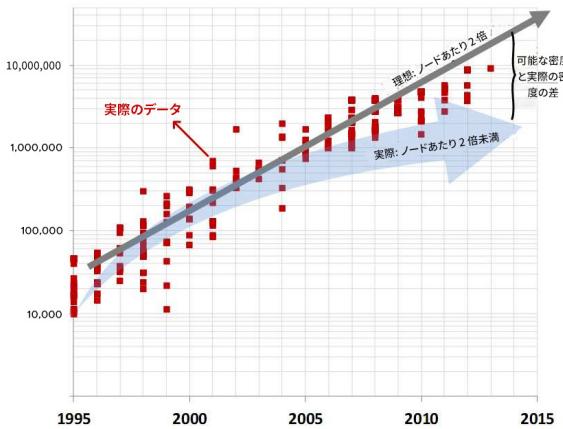


図 2: 拡大する EDA と設計の能力差

現に必要なコードとデータ両方の管理がより複雑になるという問題がある。この問題に対処するため、多数の団体が従うべきベストプラクティスに関する提言を作成している。以下はその一例である。

再現性とは、完全なソフトウェア環境の利用を他の研究室でも可能とし、ソースコード全体を検査、変更できるようにすること [9]

Reproducible Research Standard [10] (RRS) (Stodden, 2009 年) では、科学における再現性を可能にする基本原則の一つに言及している。

科学的使用許諾の原則: 科学研究大要の普及、共有、使用、再利用における法的負担は最小限にとどめるべきであり、適用にあたっては強力で説得力のある論理的根拠が必要である。

制限付き NDA の下でのみ利用できるデータの使用、クローズドソースの独自ツール、制限付きライセンスを組み込んだハードウェア設計は、ハードウェア設計のあらゆる局面で頻繁に見られるが、この提言に直接反するものである。ハードウェア設計分野の研究開発がこうした制約のあるリソースに大きく依存していることは、Andrew Kahng が [9, 11] で提起しているとおりである。

透明性という目標を達成できるよう研究システムを後押しする、オープンソースコードやデータセットのリリースを通じた支援は、Google の取り組みが非常に成功している分野であり、Google の研究に対する理念 [2] の中でも強調されている。

研究活動の大半において、科学出版物は重要な要素であると常に考えてきたが、基礎研究プロジェクトにおいては、オープンソースコードのリリースや新しいデータセットが特に有効なものとなる。これらは、研究を評価し、その状況を把握する際に役立つだけでなく、新しいコラボレーションを刺激するものもある。

完全オープンソースのコードやデータセットのリリースが有効な例として、Google の機械学習分野での取り組みが挙げられる。2015 年、Google は TensorFlow [12] のオープンソースコードをリリースした。その理由を、Google の極めて重要な発展の大半を支えた主要なソフトウェアエンジニアで、現在は Google の研究責任者である Jeff Dean が以下のように述べている。

私たちが期待しているのは、コミュニティがこれをさまざまな種類の機械学習アルゴリズムを表現する優れた方法として採用し、さらに、さまざまな興味深い方法により [TensorFlow] の構築と改善に貢献することである [13]

以来、Google はオープンにした TensorFlow の開発を加速させ続け、それと合わせて機械学習分野で利用する多数のオープンデータセットをリリースしてきた。TensorFlow は、営利企業が [9] の提言と同様の結論に達し、

さまざまな分野にわたる広範な研究の再現性を円滑に支える、オープンソースの有用なツールとインフラの研究開発に投資を行っている例である

この種の投資は Google だけでなく Facebook 社、Nvidia 社、Microsoft 社など多数の企業によっても行われ、当該分野での研究が爆発的に増加した [14]。

TensorFlow の分野における Google の取り組みは、Google の研究理念のもう一つの重要な側面、つまり、研究と開発チームを結びつけ、従来の技術移転プロセスの最小化または排除を目指すという点にも光を当てている [2]。Google は、研究者が開発に使用してきた技術を利用し、機械学習技術に基づくプロダクトを数多くリリースしてきた。このような高速な技術移転、および研究と開発チームの統合はどちらも、[11, 15] で強調されているように、ハードウェア設計業界が実現に苦労している分野である。

RISC-V は、前述のデータセットとコードをオープン化するトレンドの成功を雄弁に物語る、もう一つの例である [16]。多くの設計者が、命令セットアーキテクチャ (ISA) の独占的なライセンスの制約から解放された新しい実装の定義に貢献し、自分以外の設計者の作業を基にさらなるビルドを進めている。その一方で、研究分野から産業界への直接的な技術移転も急速に進んでいる。オープン性という共通の基盤ができたことで、オープンソースソフトウェアでの RISC-V アーキテクチャのサポートが、目立った阻害要因なしに実現するようになった。

また、機械学習と RISC-V の例は、研究のあらゆる側面がソフトウェア開発との関連を強めていることを示しており、これは私たちが発見した障害へのアプローチに反映されている。2016 年の米国におけるソフトウェアエンジニアの数は 80 万人であるのに対し、ハードウェアエンジニアはわずか 7 万人である。今後 10 年間で、ソフトウェアエンジニアは 30% 増加すると予想されているのに対し、ハードウェアエンジニアの増加は 5.4% にとどまる予想されている [17]。今後、ハードウェアの設計において、ソフトウェアおよびソフトウェアエンジニアが重要な役割を果たす必要があることは明らかである。本稿の著者は、図 3 に示すように、ハードウェア設計を 3 つの主要な構成要素に分け、重複と相互関連が見られる研究分野を示している。EDA ツールと TensorFlow、PDK データと ML データセット、IP とライブラリは、ML の研究分野として類似性がある。エコシステムの成功には、3 つの分野すべてが関わって大きな変化を起こすことが求められ、私たちはそれぞれの構成要素にある障害を発見し、これを取り除く取り組みを実施している。

第 3 項では、製造可能なオープンソース PDK をリリースする取り組みについて説明する。高品質でオープンに共有できるデータセットの欠如が ML 分野の発展を妨げているのと同様に、オープンソース PDK の欠如が他のすべての側面を妨げている。



図 3: PDK、EDA、IP ライブライはハードウェア設計エコシステムの 3 大構成要素

第 4 項と第 5 項では、EDA ツールのエコシステムを改善する取り組みについて説明する。この取り組みは、Google の TensorFlow と多くの類似点があるが、主に OpenROAD などの既存プロジェクトとの連携とその強化に重点を置いている。

第 6 項では、IP とライブライの重大な欠点を修正し、IP とライブライを生成するための新しいアプローチを可能にする取り組みについて説明する。ここでも、OpenRAM、BAG、FASoC などの既存の取り組みとの連携やその強化に重点を置いている。

3 障害 - 製造可能なプロセス設計キット

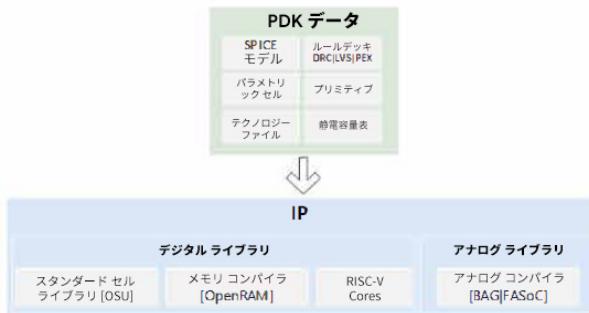


図 4: オープンソース化されたプロセス設計キットデータ: ASIC 設計に必要なビルディング ブロック ライブライの作成を可能にする、最初の支援レイヤ

ハードウェア設計では、基本的な構成要素はファウンドリからプロセス設計キット (PDK) として提供される (図 4 参照)。PDK データは、ハードウェア設計研究のあらゆる側面に不可欠なものである。PDK データには制限付きライセンスが付与されるため、オープンソースの RTL や EDA ツールを使用しても、テープアウトした最終設計は一般に公開されない [18]。図 5 に示すように、PDK データは、入力設計仕様から製造可能な設計への変換に必要なほぼすべてのステップに不可欠である。図 5 は、使用されている IP と RTL ライブライがオープンであっても、電子設計自動化 (EDA) ツールの出力データが PDK と密接に関連していることを詳細に示している。したがって、オープンデータがなければ、オープンソースの RTL と EDA ツールの両方の開発に直接影響が及ぶこととなる。EDA ソフトウェアは、このコンテキストでなければ、正確性をテストおよび評価できないのである。このように、PDK データの欠如は、オープンソースの EDA ツールにとって、公開された継続的インテグレーション (CI) システムがないことを意味する。これにより生産性が著しく阻害され、ツー

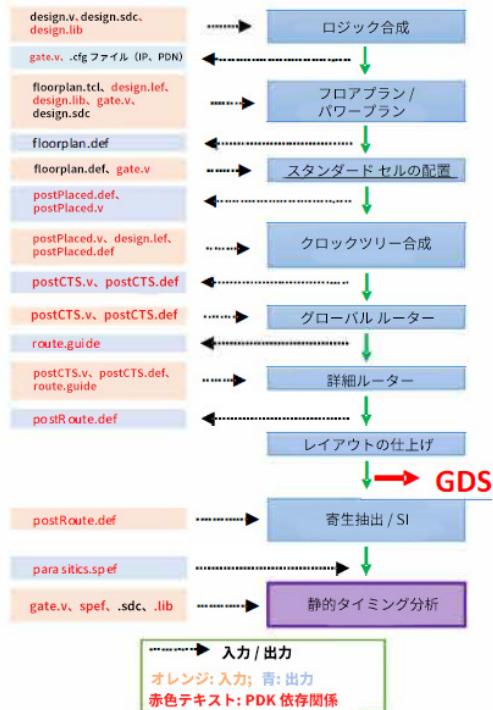


図 5: 典型的な EDA ツールフローのステージと、関連する PDK の影響 (赤字で表示)

ルの開発に貢献できる人の数が大幅に減少する。最新のソフトウェア開発手法では、継続的インテグレーションが多用されるため、この制約により多くの優秀なソフトウェア エンジニアが追いやられてしまうのである。

Verilog や VHDL などの言語の開発により、デジタル RTL は PDK から独立して開発できるようになったが、RTL の変更によって結果が改善したのかを確認するには、RTL をスタンダードセル ライブライ、ひいては PDK にリンクしなければならないことに変わりはない。実際、RTL の変更は、不明瞭な形で最終結果に大きな影響を与えることがある [19]。一方、アナログ設計に相当する高水準記述言語が利用できないため、現状では PDK データにアクセスせずにアナログ IP を開発することはほぼ不可能である。こうした点は、成長かつ成功するエコシステムにとって、完全オープンソースのコードとデータセットが重要なことを一層際立たせている。この点は第 6 項でも示されている。こうしたオープン性は、基本的な構成要素が門外不出となっている限り、実現できないことである。

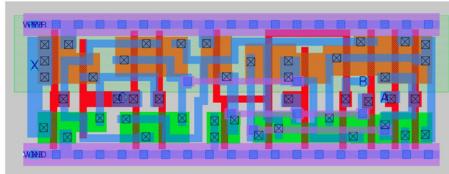


図 6: MagicLayout Editor で視覚化された SKY130 ライブ ラリのスタンダードセルレイアウト

徹底的な評価の結果、製造可能なオープン PDK が存在しないことが判明した。存在する PDK はオープンガイドラインを遵守しておらず、実際に製造可能なデータと合致しない。この点についても、Andrew Kahng が [11, 15] で取り上げている。こうした制限に対処するため、Google は SkyWater Technology Foundry 社と協力して、130 nm プロセスノード (SKY130) をオープンにリリースしている [20]。このノードは 20 年前のも のだが、アナログ設計に適した特性を有するため、非常に安価であり、現在でも本番製造として重宝されていることが確認された。SKY130 ノードは、ローカル相互接続、Silicon-Oxide-Nitride-Oxide-Silicon (SONOS) 機能、MiM コンデンサーなど、多彩な機能と上質なオプションを備え、130 nm の水準を押し上げている。これにより研究開発の機会が広がっている。また、SkyWater Technology Foundry 社は、Nb、Ge、V2O5、カーボン ナノチューブなどの特殊な材料や技巧を取り入れるなどし、プロセスをさらにカスタマイズして量産可能にすることを専門としている。これは、研究から量産への高速技術移転を促進するという目標に合致している。

The SKY130 PDK は GitHub から入手可能であり、秘密保持契約 (NDA) は一切ない。制約の緩い Apache 2.0 ライセンスが使用され、プラットフォームのコンテンツは、あらゆるソフトウェアと同様に扱われる。表 1 に示すように、SKY130 PDK は、オープンソースとプロプライエタリの両方の設計フローに必要なすべてのファイル、アナログ設計を作成するためのプリミティブセルライブラリとモデル、SPICE モデルに加え、デザインルールチェック (DRC)、レイアウトと回路図 (LVS)、およびサインオフ寄生抽出を検証するルールデッキをサポートする。ハードウェア設計者とソフトウェア開発者は、従来の制約から解放され、新しい IP やツールを共有、複製し、以前のプロジェクトの上に段階的に構築することが可能となる。

PDK ダイアグラム	ルールデッキ	行動モデル	SPICE モデル	PCell	テクノロジーファイル
	DRC LVS PEX	アナログ デジタル	Monte Carlo プロセス コーナー	トランジスタ コンデンサー インダクタ 寄生	LEF 静電容量表 レイヤマッピング

表 1: SKY130 プロセス設計キットの内容

SKY130 PDK には、表 2 に示すように、低電力、マルチスピード、高密度、高電圧などさまざまなタイプを対象とするスタンダードセルライブラリの初期セットが付属している。高密度スタンダードセルレイアウトの一例を図 6 に示す。SKY130 PDK は、カスタマイズされた SRAM やフラッシュ ビルドスペースなど、アナログ IP の開発に必要なリソースを提供するが、アナログ IP は含まれない。この問題にどのように取り組んでいるかについては、第 5 項で説明する。

この PDK が、ファウンドリ提供のセルに限定されない新たな選択肢として、誰もがまったく新しいスタンダードセルを開発できるほどに高い完成度を有することを証明するため、オクラホマ州立大学 (OSU) に新しいスタンダードセルを作成してテストするための助成金が提供されている。

デジタル スタンダードセル	メモリ コンバイラ	ビルドスペース	IO と周辺機器
高密度 高中低速 低リード 高電圧 省電力	OpenRAM	SRAM ピットセル DRC 例外 SONOS Flash ピットセル	デジタル/アナログ IO

表 2: SKY130 のライブラリとデジタルデザインスペース

最後に、SKY130 テクノロジーへのアクセスに関する障害をさらに軽減し、特にこのプロセスノードをターゲットとした新しい RTL と IP のリリースを奨励するために、Google は Efabless 社と SkyWater 社に資金提供を行い、完全無料のオープンソース シャトル プログラム [20] を立ち上げた。完全なオープンソース 設計を対象に、2 年間にわたって複数の実行をカバーする予定である。学術団体、趣味同好会系団体、営利団体によって作成されたオープンソース設計であれば、このプログラムに参加できる。シャトルランの IP はすべて利用できるため、設計者は複数回のシャトルランを行うことで、1 回のシャトルランで得られた良好な結果を次回以降のシャトルランで活用することが可能となる。幾度も試行を重ねることが漸進的な改善につながると期待されるため、これは重要なポイントである。ML 分野での機械学習データセットのリリースの例で述べたように、Google は、PDK 情報をオープンソースドメインにリリースすることが他のファウンドリ企業の間でも一般化することを期待している。

4 障害 - ツールの配置配線



図 7: Google での週あたりの commit 数

完全にオープンで再現可能なハードウェア設計エコシステムを構築するうえでもう一つ問題となるのが、ハードウェア設計の作成に使用するツールである。多くの研究者は、一般的に使用されている主要ベンダーの独自ツールを自由に利用できるが、そのツールの変更または適応は行えない。また、オープンソースの RTL やオープンソースの PDK データでツールを使用する場合でも、こうしたツールのライセンスにより、生成された生の結果やデータを配布できないことが多く、検証をさらに困難にしている。研究者に提供されるライセンスには、そのアプリケーションに関心を持つ企業への直接的な技術移転を禁止する条項がほぼ例外なく盛り込まれている。このため、ツールの改善を支援することに、他の営利企業も少なからず金銭的インセンティブを持っていたとしても、開発チームを参画させることができないでいる。このようにツールを自由に変更、改良できることによる弊害は、共通点のある他の分野で Google によってすでに実証されている。たとえば、GPU コンパイラの分野では、最大 51% のパフォーマンス向上が可能であることが実証されている [21]。

こうした障害の重大性をさらに強調するものとして、DARPA は ERI イニシアチブの中で、EDA のパフォーマンス ギャップを解消し、ムーアの法則の終焉に対処するには、ハードウェアの設計製作や設計反復に要する時間を劇的に短縮する必要がある

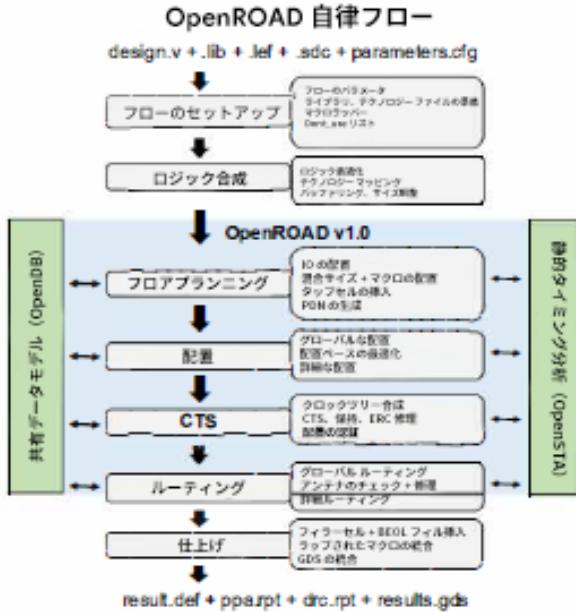


図 8: OpenROAD 自律フロー

と指摘した [22]。ASIC の配置配線への ML 適用に取り組んでいる別のグループも [23]、EDA ツールの実行時間を大幅に短縮することで、ハードウェア設計の生産性を劇的に向上できると結論づけている。

EDA ツールのランタイムを最適化する必要性は、DARPA の IDEA プログラムにおいて、課題である「シリコンコンパイア」の製作を阻む大きな障害として認識されていた [24]。人間参加型でない、24 時間という積極的なターンアラウンド時間の実現には、複数の研究チームによる EDA フローのあらゆる部分に大幅な変更を加える必要がある [15]。これは、既存の EDA ツールの現状では容易なことではなく、Andrew Kahng を中心とした OpenROAD プロジェクト [25] の創設につながった。OpenROAD プロジェクトは、カスタマイズが容易なオープンソースのコアベースを構築し、ハードウェアの問題を扱いやさしいサブ問題に積極的に分解し、並列および分散探索ソリューションを適用して問題に取り組むことを目的としている。

問題を複数の小さなチャンクに分割し、信頼性の低いコモディティ ハードウェアのクラスタ上で大規模に実行できるようするために、Google は Borg [26]、Bigtable [27]、MapReduce [28] などのテクノロジーを開発し、この分野を牽引している。また、自社のソフトウェア エンジニアの生産性を高めるために、以前は限界と考えられていたレベルを超えてツールをスケーリングできることを実証した [29]。また、こうした進歩により、Google はソフトウェア エコシステムを継続的に進化させ、図 7 で示すように開発者の生産性を向上させている。OpenROAD プロジェクトが Google の中核的な強みとこれまでの取り組みにいかに密接に関わっているかを考えると、OpenROAD プロジェクトを、専門知識やコンピューティング リソース、当該プロジェクトへの研究助成金で支援するという選択は容易なことだったといえる。

この 2 年間で、OpenROAD プロジェクトは主要なマイルストーンを達成できた。2020 年の Electronic Resurgence Initiative (ERI) サミットで、12 nm finFET テクノロジーを使用するテーブアウト データベースの実現が発表された。これには、49 個の SRAM マクロを含む 53 万 5,000 以上の配置可能な設計要素で、人手をかけずに DRC クリーンな結果を生成する必要があった。



図 9: sv-tests ダッシュボードのスクリーンショット

Google は Efabless 社に資金提供を行って OpenROAD プロジェクトで連携して、オープンソース PDK と組み合わせたこのテーブアウト データベースを OpenLANE フローに使用し、このフローを既存のオープンソース RTL でテストして、striVe SoC ファミリを構成する複数のオープンソース設計をテーブアウトしている。これはすべて、完全に DRC クリーンで人手を介さないテーブアウトを目標としている。すでに 3 種類の設計が完成しており、今後も製作が予定されている [18]。

5 障害 - 既存の SystemVerilog エコシステム

OpenROAD プロジェクトで複雑な ASIC 設計の配置配線に成功する一方で、Google は EDA ツール分野において、既存のオープン IP エコシステムの大部分をオープンソース ツールで使用できないという別の障害を特定した。デジタル設計の大部分は、プロプライエタリかオープンソースかを問わず SystemVerilog 言語で書かれている。この言語はオープンソース ツールによるサポートが不十分であることが指摘されている。まずこの指摘を検証するために、Google は Antmicro 社と共同で、SymbiFlow [31] プロジェクト下で sv-tests [30] という完全オープンソースの SystemVerilog コンプライアンス スイートを開発した。このテストスイートは、プラウザの CSS コンプライアンスを高めるために開発された CSS ACID テストと同じ原理を踏襲している。sv-tests ダッシュボードのスクリーンショットを図 9 に示す。テストスイートが順調に進んでいたため、Google はオープンソースとプロプライエタリ両方のさまざまなツールのコンプライアンスを評価でき、オープンソースのツールについては、テストスイートやツールが更新されるたびに、コンプライアンス情報を完全に公開することができた。このテストスイートに関してさまざまな人たちからの多大な貢献があり、SystemVerilog サポート関連で特定できていなかった作業も把握することができた。この情報から SystemVerilog の内部リンターおよびフォーマッタである Verible のコンプライアンスが適切である [32] ことが判明し、これをオープンソースとしてリリースすることとなった [33, 34]。Verible は、Google 社内の多くのプロジェクトや、公開されている OpenTitan プロジェクト [35] のリンターとして使用されており、Western Digital 社など、RTL 開発を行う他企業にも採用され始めている。

また、OpenROAD プロジェクトを通じて、Surelog という名の ANTLR ベースの SystemVerilog パーサーを独自に開発していた Data Model Solutions 社という小さな企業とも連携を図った。sv-tests によってパーサーの品質を正確に評価できるようになっ

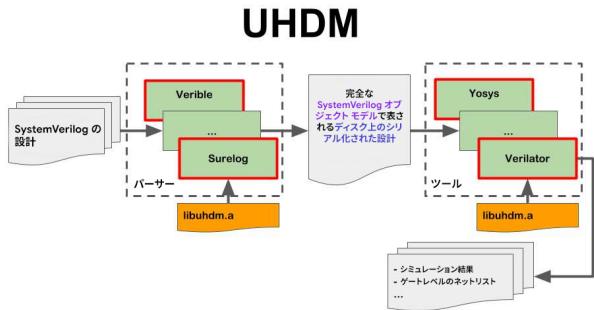


図 10: UHDM アーキテクチャのブロック図

たため、Google はこのプロジェクトに資金提供を行い、Surelog をオープンソース ライセンスでリリースすることが可能となった。Surelog は現在利用可能であり [36]、以下のような広範な機能セットを提供している。

- sv-test スイートのほぼすべてのテストにおいて、最も高い合格率を誇るツールの一つ
- マルチスレッド プリプロセッサおよびパーサー（マルチスレッド コンパイル用にファイル、モジュール、パッケージを分割）
- カスタムの lint ルール作成、および類似する他の機能を有効にする包括的な Python API

Surelog の研究を基に、Google のエンジニアは現在、Data Model Solutions 社と Antmicro 社と協力して、Universal Hardware Data Model (UHDM) プロジェクト [34, 37] を立ち上げ、Surelog パーサーを Yosys や Verilator [38, 39] などの主要なオープンソース ハードウェア ツールに統合できるようにしている（図 10 参照）。

6 障害 - IP エコシステムの欠落部分

オープンソースの IP と RTL は、1999 年から OpenCores [40] のようなオンライン コミュニティが存在するなど、長い歴史がある。Google の他のグループは、OpenTitan [35] などのプロジェクトで、すでにこの分野に貢献を行ってきた。デジタル設計は、さまざまなツール間、プロセスノード間、さらには FPGA と ASIC のテクノロジー間でさえ簡単に移植できるため、オープンソースのデジタル IP エコシステムにおいては一定の前進が見られる。RISC-V などの最近の成功や FPGA ベースのプラットフォームの大幅なコストダウンにより、オープンソースのデジタル設計への貢献は増える一方である。アナログ設計のような他の分野は残念ながら、基本的なオープンソースのアナログ IP ブロックさえ存在しない。

多くのアプリケーションにおいて、アナログ IP は最終的な ASIC 設計に多大な影響を及ぼすものである。一部がアナログ方式である組み込み SRAM は、ほぼすべての最新 IC の大部分を占めている [41]。電源の安定性はスイッチング周波数に影響し、電源効率は消費電力に直接影響する [42, 43]。モノのインターネット (IoT) 市場の予測 [44] によれば、ソフトウェア定義無線 [45] などの Big Analog Small Digital (大半がアナログ回路で小規模なデジタル回路が混載) の IC を持つミックスド シグナルシ

ステムが ASIC 設計の主流になると予想される。アナログの高速 SerDes と IO は、最新の IC 間データ転送のニーズを満たす基本である [46]。したがって、オープンソースのアナログ IP が存在しないことは、たとえば DARPA が IDEA イニシアチブ [24] で示しているように、再現性と、実際の設計への研究の適用に対する大きな障害であることは明らかである。

Efabless 社の Ravenna IC は、そのわかりやすい例であり、IC の他の部分はオープンソース IP として公開されているが、ADC、電源、SRAM ブロックなどのアナログ ブロックはファウンドリから提供され、プロプライエタリのままである。striVe SoC ファミリも、現状では SKY130 プラットフォームにアナログ IP がないため、Ravenna IC ラインと同等の機能には至っていない。こうしたアナログ IP の欠如により、LiteDRAM [47] や LiteX エコシステム [48] など、既存の高品質 FPGA ベースのオープンソース デジタル IP を SKY130 プラットフォームに移植する能力が低下することとなる。

組み込み SRAM は、ほとんどの最新の ASIC 設計で非常に重要な役割を果たしているため、SKY130 プラットフォーム、特に SRAM ビルドスペースを最大限に活用できるよう、完全オープンソースのメモリ コンパイラが必要であることがわかった。オープンソースのメモリ ジェネレータは多数あるが、Google が調査したプロジェクトのうち OpenRAM 以外は、生成プロセスの一部にクローズド ソース ツールを必要とするものであった。OpenRAM は、クローズド ソースの独自ツールだけでなく、完全オープンソースの代替品もサポートしているため [49]、助成の対象となって SKY130 プラットフォーム用のメモリ ジェネレータを構築できた。最初のテスト用 SRAM 設計は製造に回され、OpenRAM SRAM を使用することで striVe の設計サイズを大幅に削減することが可能となった（図 11 参照）。これはまだ最初のメモリ製造にすぎず、SKY130 ビルドスペースにある一部のカスタマイズされたセルを利用しているだけである。OpenRAM チームでは、さらなる最適化を計画している。

[50] とオープンソース IP のシャトル プログラム [20] により、オープンソースのアナログ IP の開発が可能になったため、この分野の活性化が期待される。アナログ IP の重要性が増していることから、Google は SKY130 プラットフォームを対象とした 2 つの異なるアナログ生成アプローチの実現のために出資を行ってきた。

BAG: Berkeley Analog Generator [51] は、設計者が複数のテクノロジーノード間で移植可能な Python ベースのジェネレータを簡単に作成できるように設計されたツールである。そのためには、ベースレベルのプリミティブ ライブラリを作成する必要があり、ジェネレータはこのライブラリ上に構築される。BAG ジェネレータのアプローチは 3 回反復され、ツールの商業化を目的とした Blue Cheetah 社というスタートアップ企業の創設も並行して行われた [52]。Blue Cheetah 社は、Intel のチップ レット スタイルの製造をターゲットとしたオープンソースの Advanced Interface Bus (AIB) バージョン 2.0 を開発し、BAG テクノロジーを活用して成功を収めたことを示した [53]。Google は、BAG に必要な基本プリミティブの作成を Blue Cheetah 社に委託し、BAG テクノロジーを利用して誰でも SKY130 プラットフォーム用のアナログ IP を開発できるようにした。しかし、BAG はまだ多くのステージで独自のツールを使用する必要があり、コミュニティで幅広く採用されるには難しい段階である。

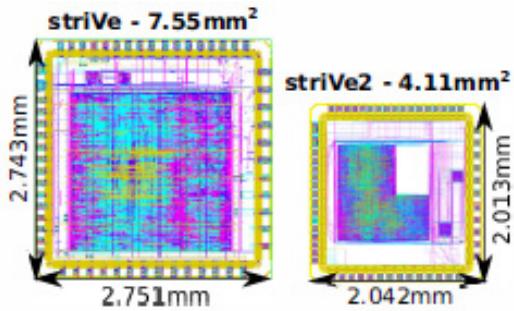


図 11: striVe および striVe2 のテストチップの縮尺図
(OpenRAM で生成されたメモリブロックの使用によるサイズの縮小を示す)

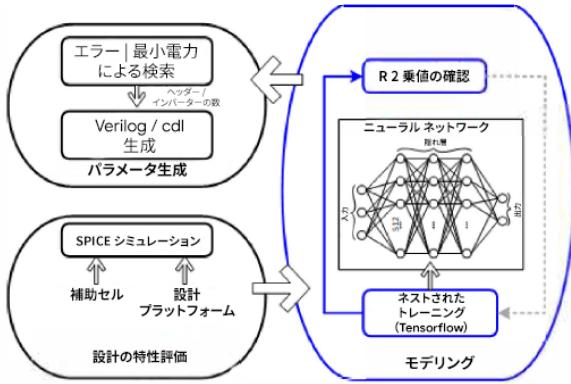


図 12: ネストされたトレーニングと、最小誤差または電力に基づくパラメータ検索を用いたニューラルネットワークの自動化フロー

FASoC: Fully Autonomous System-on-Chip (FASoC) [54] ツールは、IDEA プログラム [24] 内の DARPA 出資プロジェクトであり、主な目的は、アナログ コンパイラのニーズに対応することである。デジタル設計フローは古くから自動化が進んでいるが、ほとんどのアナログ自動化フローは、いまだに手動設計に大きく依存している。こうしたアナログ自動化ツールの欠如は、設計サイクルの長期化とコストアップにつながり、完全自律設計を実現するうえで大きなハードルとなっている。したがって、アナログ ブロックの自動生成を含むシリコン コンパイラの実現は、大きな課題ではあるものの、綿密なアナログ設計にかかる製品化までの時間に要する煩雑なコストを軽減する可能性を含んでいる。FASoC への取り組み [55] では、セルベースのアナログ設計生成手法により、さまざまなアナログ / ミックスシグナル (AMS) ブロック (PLL, LDO, DC-DC コンバータ、ADC など) を製作している。各ブロックレベル ジェネレータの主要なアナログ機能を特定し、補助セルと呼ぶこととする。特定の設計キットで必要なすべての補助セルを有するライブラリを構築することにより、標準的なデジタル合成フローでこれらのセルを使用することができ、アナログ回路設計の自動化が可能となる。このプロジェクトにより、アナログ ジェネレータの SkyWater テクノロジーへの移植、現在のクローズド ソースの EDA ツールからオープンソースの EDA フローへの置き換えを支援する。

[56] で定義されている構造のバリエーションを使用して、温度センサーが移植されており、このジェネレータは、温度範囲

(TRANGE) 、電力、不正確性の制約、最適化戦略などのユーザー入力仕様を受け取る。提案中の自律フローの初期ステップでは、ツールによりモデリング ファイルがすでに提供されているかどうかを判別する。まだ存在しない場合は、センサーの設計スペース探索の一環として、センサーの内部パラメータを検査する SPICE シミュレーションを実行することにより、新しいファイルが自動的に生成される。次に、その結果を、ペイジアン ニューラル ネットワーク (BNN) モデルのトレーニングに使用する (図 12 参照)。これにより、シミュレーションの実行時間が大幅に短縮され、ジェネレータは目的の入力仕様に最適な設計パラメータの構成を予測および決定することが可能となる。階層構造トポロジを採用することで、低消費電力や高性能など、アプリケーションに応じた最適化が実現する。ここで取り上げているジェネレータは、アナログの各機能を Verilog で記述できるため、従来のデジタルフローに対応する。図 13 の例のように、アナログ ブロックをソフトマクロとして扱い、最小限のデジタルフロー制約で配置配線できるため、この最後の機能は非常に魅力的である。

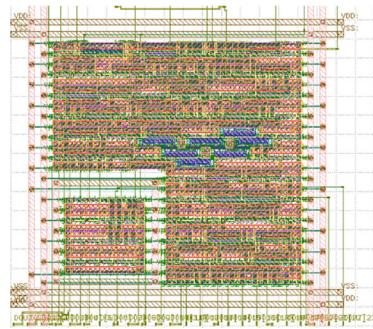


図 13: SKY130 における温度センサー自動設計生成 [FASoC] の上面図

7 結論

コンピューティング要件が急増している中、Google は増え続ける需要に対応できるハードウェア設計を可能にするため、複数の手段を検討している。こうした取り組みの一つを Tim Ansell のグループが主導している。その取り組みでは、急速に成長するハードウェア開発エコシステムの実現に主眼を置き、研究者および業界と協力して、的を絞ったアプローチを取っている。そのアプローチとは、認められている科学的なベスト プラクティスを用いて、オープンかつコラボレーション指向の研究エコシステムの構築を阻む主な障害を取り除くというものである。

知識を共有することであらゆる人の進歩を加速し、研究と開発チームをつなぐという Google Research の理念に従い、また、機械学習分野における TensorFlow を中心とした Google の取り組みが示した成功の道筋を辿るべく、Google は新しいオープンソース コードとデータセットをリリースしてきた。また、さまざまな分野にわたる幅広い研究に対して再現性を円滑に支援するため、オープンソースの有用なツールとインフラの研究開発への投資を継続している。

取り組みを通じて、ハードウェア設計に必要な 3 つの基本領域においてさまざまな障害があった。以下はその対策の一例である。

- ・製造可能なPDKのオープンソースリリース[50]と、スタンダードセル作成用PDKのユーザビリティ検証を行うためのOSUへの資金提供
 - ・OpenROADプロジェクト[25]との連携や、SystemVerilog関連ツールの開発とオープンソースリリース[32]など、ツールへの貢献
 - ・OpenRAM[49]などのコアIPに関する資金面の支援、BAG[51]やFASoC[55]などの有望かつ興味深い新たなアナログジェネレータの実現の支援
- ここに至るまでの経緯はすでに驚嘆すべきものであるが、こうした取り組みからさらに興味深い結果が得られることが今後一層期待される。

謝辞

本稿の準備および内容の精査にあたって、OpenROADチームの皆様に計り知れないほどのご協力をいただきました。心より感謝いたします。また、助言やサポートをくれたDanny BerlinとParthasarathy Ranganathanにも謝意を表します。最後に、多くの協力者とオープンソースコミュニティメンバーの皆様のおかげで、本稿を発表できる運びとなりました。ここに御礼申し上げます。

参考文献

- [1] Nathan Myhrvold. The future of software, the software industry, and windows '47. In *ACM97: The Next 50 Years of Computing*, ACM '97, New York, NY, USA, 1997. Association for Computing Machinery.
- [2] A. Spector et al. Google's hybrid approach to research. *Commun. ACM*, 55(7): 34–37, July 2012.
- [3] L. Hatton et al. The long-term growth rate of evolving software: Empirical results and implications. *Journal of Software: Evolution and Process*, 29(5):e1847, 2017.
- [4] A. Lottarini et al. Vbench: Benchmarking video transcoding in the cloud. *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '18, pp. 797–809, New York, NY, USA, 2018. Association for Computing Machinery.
- [5] Neil Thompson. The economic impact of moore's law: Evidence from when it faltered. *SSRN Electronic Journal*, 01 2017.
- [6] N. P. Jouppi et al. In-datacenter performance analysis of a tensor processing unit. 2017. URL <https://arxiv.org/pdf/1704.04760.pdf>.
- [7] Andrew B. Kahng. Reducing time and effort in ic implementation: A roadmap of challenges and solutions. *Proceedings of the 55th Annual Design Automation Conference*, DAC '18, New York, NY, USA, 2018. Association for Computing Machinery.
- [8] Roger D. Peng. Reproducible research in computational science. *Science*, 334 (6060):1226–1227, 2011.
- [9] National Academies of Sciences Engineering and Medicine. *Reproducibility and Replicability in Science*. The National Academies Press, Washington, DC, 2019. URL <https://www.nap.edu/catalog/25303/reproducibility-and-replicability-in-science>.
- [10] Victoria Stodden. Enabling reproducible research: Open licensing for scientific innovation. *Science*, 13, 03 2009.
- [11] Andrew B. Kahng. Looking into the mirror of open source. In *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1–8. IEEE, 2019.
- [12] M. Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL <https://www.tensorflow.org/> から入手可能なソフトウェア。
- [13] Google. Tensorflow: Open source machine learning, 2015. URL https://youtu.be/oZikw5k_2FM.
- [14] Y. Duan et al. Artificial intelligence for decision making in the era of big data - evolution, challenges and research agenda. *International Journal of Information Management*, 48:63 – 71, 2019. URL <http://www.sciencedirect.com/science/article/pii/S0268401219300581>.
- [15] Andrew B. Kahng. Open-source eda: If we build it, who will come? In *Proc. 28th IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, 2020.
- [16] S. Hoover. Unleashing the potential of open-source silicon. Invited talk, VSDOpen Conference, 2019.
- [17] Sokanu Interactive. The job market for computer hardware engineers in the united states, 2020. URL <https://www.careerexplorer.com/careers/computer-hardware-engineer/job-market>, <https://www.careerexplorer.com/careers/software-engineer/job-market>.
- [18] R. T. Edwards et al. Real silicon using open source eda. *IEEE Design & Test*, 2020.
- [19] P. Davide Schiavone et al. Slow and steady wins the race? a comparison of ultra-low-power risc-v cores for internet-of-things applications. In *2017 27th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, pp. 1–8, 2017.
- [20] Tim Ansell. [fossi dial-up] tim ansell - skywater pdk: Fully open source manu-facturable pdk for a 130nm process, 2020. URL <https://youtu.be/EczW2IWdnOM>.
- [21] J. Wu et al. Gpucc - an open-source gppgpu compiler. In *Proceedings of the 2016 International Symposium on Code Generation and Optimization*, pp. 105–116, New York, NY, 2016. URL <http://dl.acm.org/citation.cfm?id=2854041>.
- [22] A. Olofsson. Silicon compilers - version 2.0, 2018. URL <http://www.ispd.cc/slides/2018/k2.pdf>.
- [23] A. Mirhoseini et al. Chip placement with deep reinforcement learning, 2020.
- [24] DARPA, 2020. URL <https://www.darpa.mil/program/intelligent-design-of-electronic-assets>.
- [25] T. Ajayi et al. Toward an open-source digital flow: First learnings from the openroad project. In *Proceedings of the 56th Annual Design Automation Conference 2019*, DAC '19, New York, NY, USA, 2019. Association for Computing Machinery.
- [26] A. Verma et al. Large-scale cluster management at google with borg. In *Proceedings of the Tenth European Conference on Computer Systems*, EuroSys '15, New York, NY, USA, 2015. Association for Computing Machinery.
- [27] F. Chang et al. Bigtable: A distributed storage system for structured data. *ACM Trans. Comput. Syst.*, 26(2), June 2008.
- [28] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: A flexible data processing tool. *Commun. ACM*, 53(1):72–77, January 2010.
- [29] Rachel Potvin and Josh Levenberg. Why google stores billions of lines of code in a single repository. *Commun. ACM*, 59(7):78–87, June 2016.
- [30] SymbiFlow, 2020. URL <https://github.com/SymbiFlow/sv-tests>.
- [31] SymbiFlow, 2020. URL <https://symbiflow.github.io>.
- [32] David Fang. Systemverilog productivity tools, 2018. URL <https://woset-workshop.github.io/PDFs/2018/a10.pdf>.
- [33] Google, 2020. URL <https://github.com/google/verible>.
- [34] Tim Ansell. Google's slides for eri posh + idea integration exercise, 2020. URL <https://j.mp/eri20-goog>.
- [35] lowRISC, 2020. URL <https://opentitan.org>.
- [36] Data Model Solution, 2020. URL <https://github.com/alainmarcel/Surelog>.
- [37] Data Model Solution, 2020. URL <https://github.com/alainmarcel/UHDM>.
- [38] D. Shah et al. Yosys+nextpnr: An open source framework from verilog to bit-stream for commercial fpgas. In *2019 IEEE 27th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, pp. 1–4, 2019.
- [39] Veripool, 2020. URL <https://www.veripool.org/wiki/verilator>.
- [40] OpenCores, 2020. URL <https://opencores.org/>.
- [41] M. Khayatzadeh et al. 17.3 a reconfigurable dual-port memory with error detection and correction in 28nm fdsoi. In *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 310–312, 2016.
- [42] J. Lee et al. 19.2 a 6.4pj/cycle self-tuning cortex-m0 iot processor based on leakage-ratio measurement for energy-optimal operation across wide-range pvt variation. In *2019 IEEE International Solid-State Circuits Conference - (ISSCC)*, pp. 314–315, 2019.
- [43] M. Saligane et al. An adaptive body-biasing soc using in situ slack monitoring for runtime replica calibration. In *2018 IEEE Symposium on VLSI Circuits*, pp. 63–64, 2018.
- [44] C. Perera et al. A survey on internet of things from industrial market perspective. *IEEE Access*, 2:1660–1679, 2014.
- [45] M. Condoluci et al. Enabling the iot machine age with 5g: Machine-type multicast services for innovative real-time applications. *IEEE Access*, 4:5555–5569, 2016.
- [46] S. Abdennadher et al. At speed testing challenges and solutions for 56gbps and 112gbps pam4 serdes. In *2020 IEEE Latin-American Test Symposium (LATS)*, pp. 1–5, 2020.
- [47] John Sully. Litedram vs mig - a viable alternative. ORCONF, 2018. URL <https://j.mp/or18-litedram>.
- [48] F. Kermarrec et al. Litex: an open-source soc builder and library based on migen python dsl, 2020.
- [49] M. R. Guthaus et al. Openram: An open-source memory compiler. In *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1–6, 2016.
- [50] Google, 2020. URL <https://github.com/google/skywater-pdk>.
- [51] J. Crossley et al. Bag: A designer-oriented integrated framework for the development of amc circuit generators. In *2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 74–81, 2013.
- [52] Inc. Blue Cheetah Analog Design. Blue cheetah analog design, 2020. URL <https://www.bcanalog.com/>.
- [53] K. Settaluri and D. Kehlet. Chiplet integration: Tools, methodology, requirement, infrastructure. Tutorial 10 DAC, 2020.
- [54] FASoC, 2020. URL <https://github.com/idea-fasoc>.
- [55] T. Ajayi et al. An open-source framework for autonomous soc design with analog block generation. In *2020 IFIP/IEEE 28th International Conference on Very Large Scale Integration (VLSI-SoC)*, 2020.
- [56] M. Saligane et al. All-digital soc thermal sensor using on-chip high order temperature curvature correction. In *2015 IEEE Custom Integrated Circuits Conference (CICC)*, pp. 1–4, 2015.