WP492 (v1.0.1) 2017 年 6 月 13 日

ザイリンクス All Programmable デバイス: 演算負荷の大きなシステムに最適な優れたプラットフォーム

著者: Cathal Murphy、Yao Fu

ザイリンクスの All Programmable FPGA/SoC は、演算負荷の高いワークロードに対応する、優れた効率、費用対効果、最小レイテンシ、設計の柔軟性、将来性を備えたコンピューティング プラットフォームです。

概要

データ処理への需要はますます増加しており、これに対応すべく将来のシステムにはコンピューティング機能の大胆な変革が求められています。x86 系プロセッサなどの従来のソリューションでは、効率と費用対効果に優れた方法で必要な処理能力を発揮できず、システム設計者は新しいコンピューティング プラットフォームを模索しています。

そこで、将来の要件に対応するコンピューティング プラットフォームとして、FPGA と GPU が注目されています。

このホワイト ペーパーでは、新しい時代に必要なコンピューティングの効率と柔軟性という観点から、GPU とザイリンクスの FPGA/SoC デバイスを比較分析します。

はじめに

クラウド データ センター (DC) や自動運転車などに用いられるシステムには、拡大し続けるワークロードとその基盤となるアルゴリズムの進化をサポートするするために演算性能の大胆な変革が求められています [参照 1]。たとえば、ビッグ データ分析、機械学習、ビジョン処理、ゲノム科学、先進運転支援システム (ADAS) センサー フュージョンなどのワークロードを処理するには、x86 系プロセッサなど既存のシステムの限界を超えて、費用対効果と効率に優れた方法で処理能力を向上させる必要があります。

システム設計者は、これらの要件に対応する新たなコンピューティング プラットフォームを探し求めています。このようなプラットフォームには、既存のインフラストラクチャに容易に統合でき、幅広いワークロードとアルゴリズムの進化に適合する高い柔軟性が必要です。さらに、これらのシステムの多くは、自動運転車などのリアルタイム システムに必要な高速応答を実現し、確定的な低レイテンシで動作する必要があります。

グラフィックス プロセッシング ユニット (GPU) のベンダー各社は、主に機械学習のトレーニングにおける高性能コンピューティング (HPC) の成功を背景に、新しい時代に最適なコンピューティング プラットフォームとして GPU を積極的に位置付けてきました。この取り組みの中で、GPU ベンダーは、機械学習の推論ワークロードをターゲットとしてアーキテクチャを変更してきました。

その一方で、GPU アーキテクチャの根本的な制限は常に見落とされています。GPU が費用対効果と効率に優れた方法でシステム レベルでの必要な処理能力を提供できるかどうかは、これらの制限の影響を大きく受けます。たとえば、クラウド DC システムでは、各種のワークロードに対する需要が時間帯によって大きく変動します。しかも、ワークロードの基盤となるアルゴリズムは急速に進化しています。GPU アーキテクチャのこういった制限のために、現在のワークロードと将来の進化したワークロードの多くを適切に GPU に割り当てることが難しくなり、ハードウェアのアイドル状態や非効率性を招きます。「GPU アーキテクチャの制限」では、これらの制限について詳しく説明します。

一方、ザイリンクスの FPGA/SoC は、将来のシステム要件に対応する理想的な選択肢としての特長を備えています。これら独自の特長には、次が挙げられます。

- あらゆるデータ タイプに対して非常に高い処理能力と効率を発揮
- 非常に高い柔軟性を備え、さまざまなワークロードで最大限の処理能力と効率を実現
- I/O の柔軟性に優れ、システムへの容易な統合と効率の向上が可能
- 大容量オンチップ メモリ キャッシュにより、効率の向上と低レイテンシを実現

「ザイリンクス FPGA/SoC 独自の利点」では、ザイリンクス アーキテクチャの利点について、GPU のアーキテクチャおよびその制限と比較しながら説明します。

GPU の起源とターゲット ワークロード

GPU の起源は PC の時代にさかのぼります。NVidia 社は 1999 年に世界初の GPU を発表したと主張していますが、多くのグラフィックス デバイスはそれ以前から存在していました [参照 2]。GPU は、グラフィックス タスク (シェーディングやピクセル アレイ変換など) を CPU からオフロードして高速化することを目的としてゼロから設計されています。そのため、GPU のアーキテクチャは高度な並列性と高スループットを実現します [参照 3]。基本的に、GPU の主な目的は視覚的表示装置 (VDU) 用の高画質イメージのレンダリングです。

長年にわたり、大量のマトリックス数値演算を必要とする医療用画像処理といった大規模並列処理が必要でメモリ負荷が大きい少数の非グラフィックス ワークロードは、CPU ではなく GPU にインプリメントすることで、GPU の高い処理能力を活用してきました。GPU ベンダーは、GPU の市場を非グラフィックス アプリケーションへと広げる機会を認識して、OpenCL などの非グラフィックス ベースの GPU 用プログラミング言語を開発しました。これらのプログラミング言語により、GPU は汎用 GPU (GPGPU) へと進化しています。

機械学習

近年、GPU インプリメンテーションに割り当てることが効果的なワークロードの 1 つに、機械学習のトレーニングが挙げられています。GPU を利用することで、ディープ ニューラル ネットワークのトレーニング時間は大幅に短縮されます。

GPU ベンダーは、機械学習のトレーニングにおける成功を基盤として、GPU の市場を機械学習の推論 (トレーニング済みのニューラル ネットワークの展開) にも広げようとしています。機械学習アルゴリズムの進化と、有効ビット数を減らすトレンドに対応して、GPU ベンダーは GPU のアーキテクチャを変更しています。その一例は、NVidia 社が Tesla P4 製品で INT8 をサポートしたことです。しかし現在、バイナリやターナリなど、さらに有効ビット数を減らす研究が進められています [参照 4]。GPU ユーザーがこのような機械学習やその他の分野における進化を取り入れるには、今後リリースされる新しいハードウェアを購入する必要があります。一方、このホワイト ペーパーの後半で説明するように、ザイリンクスの FPGA/SoC は特有の柔軟性を備えており、ユーザーの時間と費用のロスはありません。

機械学習を基盤として、GPU ベンダーは、コンピューティングの新しい時代に最適なプラットフォームとしての地位を確立しようとしています。しかし、GPU が将来のシステムにどの程度適しているかを判断するには、GPU アーキテクチャの多くの制限とシステム要件の長期的な進化を考慮に入れた、より包括的なシステム レベルの分析が必要です。

GPU アーキテクチャの制限

ここでは、標準的な GPU アーキテクチャについてより詳しく説明し、GPU アーキテクチャの制限が各種のアルゴリズムおよびワークロードに与える影響を解説します。

SIMT ALU アレイによる並列処理

図 1 に、GPU の標準的なブロック図を示します。汎用 GPU の中核にあるのは、大規模アレイの論理演算ユニット (ALU)、すなわちコアです。これらの ALU は、一般的に、単一命令複数データ (SIMD) に類似の単一命令複数スレッド (SIMT) 方式で動作します。

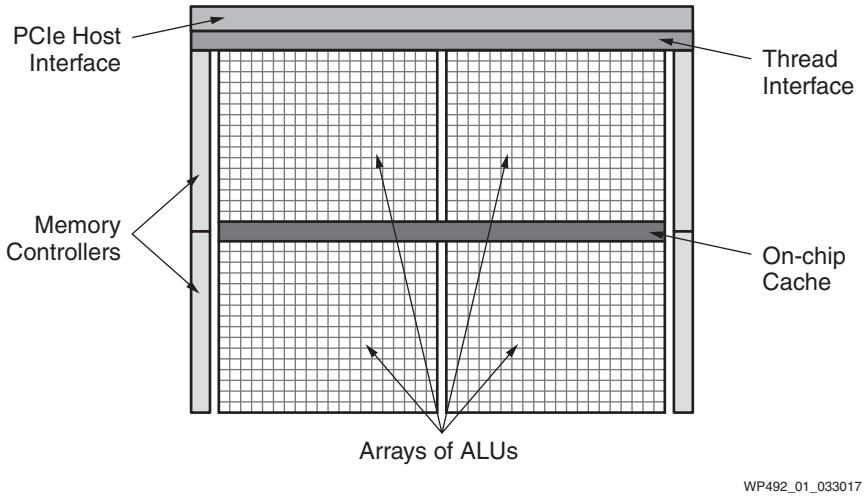

図 1: GPU のブロック図

GPU の基本原理として、ワークロードは数千の並列スレッドに分割されます。ALU のアイドル状態を防ぐには、非常に多数の GPU スレッドが必要です。これらのスレッドは、ALU の複数のグループが同じ (単一の) 命令を並列で実行するようにスケジューリングされます。コアのリソースの大部分は同じグループ内のほかのコアと共有されるため、GPU ベンダーは SIMT を利用して、CPU よりも実装面積と電力効率に優れたインプリメンテーションを実現できます。

ただし、この大規模並列処理アーキテクチャに効率的に割り当てることができるワークロード (またはワークロードの一部) は限られています [参照 5]。ワークロードを構成する複数のスレッドに十分な共通性や並列性がない場合 (たとえば、シーケンシャル ワークロードまたは中規模並列処理ワークロード)、ALU はアイドル状態になり、演算処理の効率が低下します。また、ワークロードを構成するスレッドは ALU の使用率を最大化するようにスケジューリングされるため、追加のレイテンシが発生します。NVidia 社の Volta アーキテクチャのように独立したスレッド スケジューリング機能を使用する場合でも、基盤となるアーキテクチャは SIMT のままであり、大規模並列処理ワークロードでなければ効率は上がりません。

シーケンシャル ワークロード、中規模並列処理ワークロード、またはまばら (スパース) なワークロードの場合、GPU の処理能力と効率は、CPU を下回ることもあります [参照 6]。たとえば、ゼロでない要素の数が少ないスパース マトリックス数値演算を GPU に実装した場合、GPU のパフォーマンスと効率は CPU と同等またはそれ以下になります [参照 7][参照 8]。興味深いことに、たたみ込みニューラル ネットワークの大量の冗長性を利用する手段として、多くの研究者によってスパースなたたみ込みニューラル ネットワークの研究が進められています [参照 9]。このトレンドは、機械学習の推論における GPU の問題点を浮き彫りにします。スパース マトリックス数値演算は、ビッグ データ分析の主要な要素でもあります [参照 10]。

大規模の並列処理タスクを含むワークロードには、シーケンシャルな要素や中規模の並列性を持つ要素も多少含まれていることがほとんどです。したがって、システム パフォーマンスの要件を満たすには、GPU と CPU のハイブリッド システムが必要です [参照 11]。ハイエンドな CPU の必要性は、プラットフォームの効率と費用対効果に明らかに影響を与えます。また、CPU と GPU 間に必要な通信のために、システムのボトルネックが発生する可能性があります。

SIMT/GPU アーキテクチャのもう 1 つの問題は、固定された命令セットとデータ タイプしかサポートしないため、ALU の機能が制約されることです。

限られたデータ タイプ精度のサポート

システム設計者は、認識精度を大きく低下させずにコンピューティング効率を大幅に高めるための手段として、演算の有効ビット数を減らすことを検討しています [参照 12][参照 13][参照 14]。機械学習の推論は、有効ビット数を減らすトレンドをリードする形で、まず FP16 へ、さらに INT16 および INT8 へと移行しています。さらに有効ビット数を減らす研究も進められ、現在はバイナリまで実現されています [参照 4][参照 15]。

GPU の ALU は、通常は単精度浮動小数点 (FP32) をネイティブでサポートしており、倍精度浮動小数点 (FP64) をサポートする場合もあります。FP32 はグラフィックス ワークロードに最適な演算精度であり、FP64 は一部の HPC アプリケーションで一般に選択されます。FP32 を下回る演算精度は、通常は GPU では効率的にサポートされません。したがって、標準的な GPU 上で有効ビット数を減らしても、必要なメモリ帯域幅が削減される以外のメリットはほとんどありません。

GPU は通常は若干のバイナリ演算機能を備えていますが、ALU 1 個あたり 32 のビットごとの演算のみです。32 ビットのバイナリ演算は複雑であり、大きな実装面積が必要です。2 値化ニューラル ネットワークのアルゴリズムでは、XNOR 演算に続いてポピュレーション カウントを実行する必要があります。NVidia 社の GPU は 4 サイクルごとにポピュレーション カウント操作を実行しますが、これは利用可能なバイナリ処理能力に大きな影響を与えます [参照 18]。

図 2 に示すように、機械学習の推論の進化に直面した GPU ベンダーは、必要に応じてシリコンを変更し、有効ビット数を減らした、限られた数のデータ タイプ (FP16 および INT8 など) をサポートしてきました。たとえば、NVidia 社の Tesla P4 および P40 カード上の GPU は INT8 をサポートし、1 つの ALU/Cuda コアあたり 4 つの INT8 演算を実行できます。

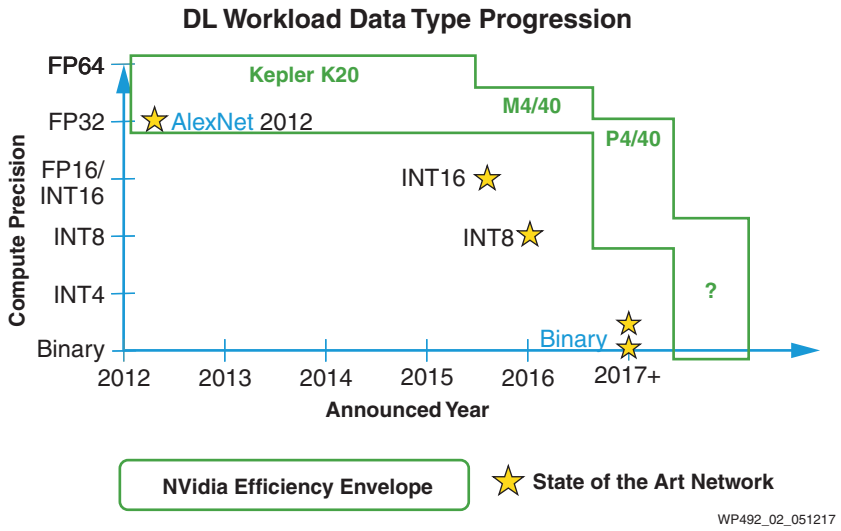

図 2: NVidia 製品の減精度演算のサポート

しかし、NVidia 社が Tesla P40 上での GoogLeNet v1 の推論について公表した機械学習推論のベンチマークによると、INT8 インプリメンテーションの効率は、FP32 インプリメンテーションの 3 倍までしか向上していません。この例は、GPU アーキテクチャで有効ビット数をぎりぎりまで減らして効率的な結果を得ることの難しさを示しています [参照 16]。

機械学習やその他のワークロードが減精度演算と可変精度演算へ移行するのに伴い、GPU ベンダーは新製品を次々と市場に投入しなければなりません。GPU の既存ユーザーがこの分野における進化の恩恵を受けるには、既存のプラットフォームをアップグレードしていく必要があります。

ソフトウェア定義のデータパスと柔軟性に欠けるメモリ階層

CPU と同様に、GPU 内のデータ フローはソフトウェアによって定義され、柔軟性に欠ける複雑なメモリ階層によって制約されます [参照 17]。図 3 に、標準的な GPU のメモリ階層を示します。個々のスレッドはレジスタ ファイルに専用のメモリ空間を持ち、スレッドのローカル変数を格納します。同じブロック内の少数のスレッドとのやり取りには、共有メモリを使用できます。ブロック外のすべてのスレッドとのやり取りには、グローバル メモリまたはオフチップ メモリを使用します [参照 18]。

図 3 に示すように、データがメモリ階層内をレジスタ ファイルからグローバル メモリへ移動すると、メモリ アクセスに関連する消費電力は 100 倍以上、レイテンシは 80 倍以上増加します [参照 15][参照 17][参照 19]。さらに、メモリ アクセスの競合が避けられないため、レイテンシの増加と ALU のアイドル状態が発生し、処理能力と効率が低下します。

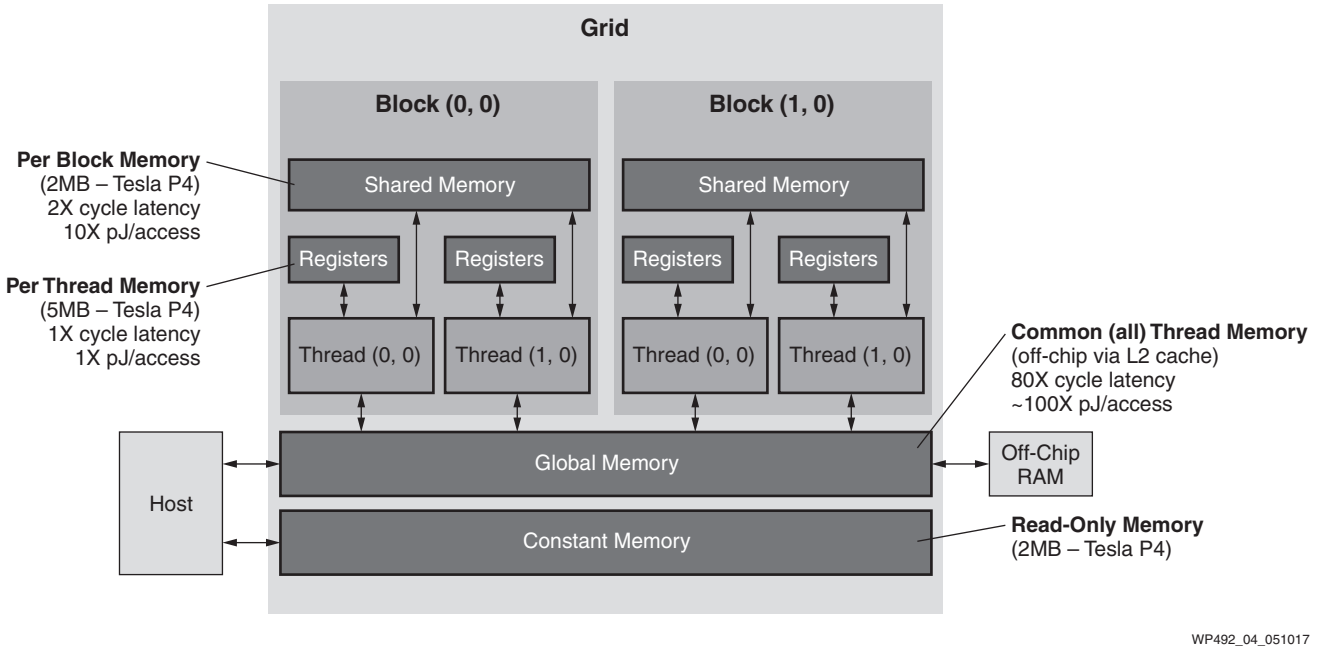

図 3: 標準的な GPU のメモリ階層

したがって、GPU が潜在的な処理能力と効率を発揮するには、ワークロードのデータ フローが GPU のメモリ階層に正確に割り当てられている必要があります。しかし、実際のワークロードのほとんどは、GPU に効率的に割り当てるにはデータの局所性が不十分です。このようなワークロードを GPU に実装した場合、実現可能な処理能力と効率は大幅に低下し、ソリューションのレイテンシは増加します [参照 19][参照 20]。

機械学習の推論の例は、このようなデータ フローの制限を定量的に明らかにしています。GPU で効率的なソリューションを実現するには、大量のバッチ処理 (バッチ = 128 など) を実行する必要がありますが、これはレイテンシを増加させます。最終的に、バッチ処理はレイテンシを犠牲にして機械学習の処理を局所化します [参照 21]。この明らかな影響は、NVidia P40 で GoogLeNet v1 の推論を実行したベンチマークに見られます。GoogLeNet v1 の場合、ネットワークは P40 のメモリ帯域幅のために GPU の処理能力に依存します。したがって、バッチ処理に関連してメモリ帯域幅が削減されても、それほど効果はありません。しかし、P40 で GPU の理論的性能の 50% を発揮するにはバッチ = 128 で実行する必要があり、システムに大きなレイテンシが発生します [参照 16]。

場合によっては、CPU によってデータを前処理し、GPU の SIMT アーキテクチャとメモリ階層へのワークロードの割り当てを助けることも可能ですが、CPU の処理能力と消費電力のコストがかかり、GPU のメリットが失われます [参照 7]。

限られた I/O オプション

「GPU の起源とターゲット ワークロード」で説明したように、GPU はコプロセッサとして設計されています。ホストとの通信を容易にするために、GPU は従来、ハード PCIe® インターフェイスと数個のオフチップ DRAM インターフェイス (GDDR5 など) のみを備えていました。最近の世代の GPU の一部は、GPU 間の通信を可能にするハード インターフェイスを備えています。この場合も、ネットワークとのインターフェイスおよび GPU へのジョブの割り当てのために CPU が必要ですが、システムの消費電力が増加する上、PCIe を介して利用可能な帯域幅が限られているためにボトルネックが発生します。たとえば、NVidia 社の Tesla P40 は PCIe 3.0 x16 をサポートしますが、16GB/s の帯域幅しか得られません。

GPU ベンダーは、GPU の処理能力と ARM® プロセッサを統合し、HDMI、MIPI、SIP、CAN、基本的なイーサネットなどの一般的な車載用ペリフェラルを搭載した、NVidia Tegra X1 などの小型 SoC の開発を始めています。しかし、これらのデバイスは大きな処理能力を持っていないため、必要な処理能力を得るためには追加のディスクリート GPU が必要になりますが、ディスクリート GPU へのインターフェイスは極めて限られています。たとえば、Tegra X1 は PCIe 2.0 x4 しかサポートしないため、これが大きなボトルネックになります。また、追加の SoC の消費電力は、プラットフォームの効率をさらに低下させます。

オンチップ メモリ リソース

オフチップ メモリは、レイテンシ、効率、スループットによる悪影響に加え、ローカル/オンチップ メモリよりもはるかに小さい帯域幅しか提供しません。したがって、ワークロードがオフチップ メモリに依存する場合、オフチップ メモリのメモリ帯域幅がボトルネックになり、コンピューティング リソースがアイドル状態になって、GPU の処理能力と効率が低下します。

したがって、低レイテンシで高帯域幅の大容量オンチップ メモリを備えていることが有利です。たとえば機械学習の推論の場合、GoogLeNet を実行するには、(FP32 インプリメンテーションと仮定すると) 重み用に 27.2MB の合計メモリ フットプリントが必要ですが、これだけのオンチップ メモリを搭載した GPU は存在しないため、オフチップ メモリが必要になります [参照 22]。多くの場合、コアがアイドル状態になるのを防ぐには、高価な高帯域幅メモリ (HBM) とバッチ処理が必要になります。大容量オンチップ メモリを搭載したデバイスを選べば、HBM から発生するコストとともに、レイテンシと消費電力も削減できます。

消費電力

GPU ベンダーは、通常は消費電力が 250W 以内になるようにカードと CPU を設計し、アクティブ熱管理を利用して温度を調整しています。機械学習の推論へと市場を拡大するために、NVidia 社は、Tesla M4 や P4 など、消費電力 75W のデバイスを開発しました。しかし、消費電力が 75W であっても、システム レベルで許容される消費電力と熱の範囲を超える可能性があります。GPU の絶対消費電力の大きさは、現在も GPU の幅広い普及を妨げる障壁になっています。

機能安全

GPU の起源は、機能安全が要件にならない、消費者向けグラフィックスおよび高性能コンピューティング市場にあります。GPU ベンダーが ADAS アプリケーションをターゲットにするようになると、機能安全は優先順位の高い必須の要件となります。デバイスが ADAS アプリケーションに採用されるには、必要なレベルの機能安全認証を取得できるように、ゼロから設計する必要があります。このプロセスは GPU ベンダーにとって長期にわたる複雑な学習曲線のようなものとなり、新しいツールとデバイスが必要とされます。

ザイリンクスの FPGA の起源

1984 年、ザイリンクスは FPGA を発明し、ユーザーがほぼ限りない数の機能を 1 つのデバイスにプログラム (および再プログラム) できるようにしました。それまで、システム設計者がこれらの機能を実装する場合は、多数の汎用ディスクリート ロジック コンポーネントを使用するか、コストのかかる ASIC を構築する必要がありました [参照 23]。

30 年以上たった今も、柔軟性とプログラマビリティは、ザイリンクスの All Programmable FPGA/SoC の主な特長です。ザイリンクスは、有線および無線通信、クラウド コンピューティング、医療、自動車、産業、航空宇宙/防衛の各市場におけるさまざまな最終アプリケーションの基本的なニーズを満たす、プログラマブルなプラットフォームを提供しています。これらのアプリケーションすべてに高度な処理能力が必要で、産業オートメーションや ADAS といった多くのアプリケーションには非常に厳格なリアルタイム処理の要件が課せられます。

FPGA を使用する際のこれまでの問題として、Verilog や VHDL などのハードウェア記述言語 (HDL) を使用して FPGA をプログラムする必要がありました。近年ザイリンクスは SDSoC™ および SDAccel™ ツールを開発し、ソフトウェア開発者やシステム設計者など、より広範囲にわたるユーザーがプログラマブル デバイスの利点を活用する可能性を開きました [参照 24]。また、ザイリンクス デバイスを活用したシステムの設計期間を短縮できるように、追加のアクセラレーション スタックを開発しました [参照 25]。

ザイリンクス FPGA/SoC 独自の利点

シンプルで強力な処理能力

GPU 支持者の主張とは逆に、ザイリンクス デバイスは強力な処理能力 (たとえば、Virtex® UltraScale+™ XCVU13P FPGA で 38.3TOP/s (INT8)) を提供します。最新の NVidia Tesla P40 カードは、ベース周波数で動作させた場合、ザイリンクス ベースのソリューションとほぼ同じ 40 TOP/s (INT8) の処理能力を発揮しますが、2 倍以上の電力を消費します [参照 26]。ザイリンクス デバイスの柔軟性とオンチップ メモリは、多くのワークロードやアプリケーションで処理能力の大幅な拡大をもたらします (「All Programmable デバイスの柔軟性」および「オンチップ メモリ リソース」を参照)。

さらに、柔軟性に優れたザイリンクスのデバイスは、さまざまなビット数のデータ タイプ (FP32、INT8、バイナリ、カスタムなど) をサポートします [参照 27]。たとえば、2 値化ニューラル ネットワークの場合、ザイリンクスは 2 値化コンピューティングで 500TOPs/s という驚異的な処理能力を実現します (1 演算あたり 2.5 個の LUT とする)。これは GPU の通常の処理能力の 25 倍に相当します。有効ビット数ごとに、DSP リソースに割り当てる、プログラマブル ロジックにインプリメントする、両者を組み合わせて使用するなど、最適な手法を選択できます。こうした柔軟性により、有効ビット数をバイナリ演算まで減らしても、デバイスの処理能力と効率を維持できます。

機械学習分野の研究の多くは、処理能力、認識精度、効率の観点から見た最適な有効ビット数を対象としています [参照 28] [参照 29][参照 30][参照 31][参照 32]。特定のワークロードに最適な有効ビット数に関係なく、ザイリンクス デバイスは高い処理能力と効率を維持することができ、減精度演算への移行のメリットをフルに享受できます。

長年にわたり、FPGA ユーザーの多くは、機械学習の推論を含むさまざまなのワークロードに対して、最適なパフォーマンスを得るためにシストリック アレイ プロセッシング デザインを実装してきました [参照 33][参照 34]。ザイリンクス FPGA/SoC のユーザーが既存のザイリンクス デバイス上でこのようなワークロードに対して最大限の処理能力と効率を実現できるように、ザイリンクスはリソースを提供しています。これらのリソースには、INT8 の最適化と、ブロック RAM および UltraRAM の最も効率的なメモリ階層への DSP アレイの結合が含まれます [参照 27]。リソースの詳細は、最寄りのザイリンクス販売代理店までお問い合わせください。

興味深いことに、現在の深層学習ワークロードに利用できる処理能力と効率を向上させるために、NVidia 社は Volta アーキテクチャの Tensor Core という形で類似の機能を強化しています。しかし、深層学習ワークロードは時間とともに進化していくため、Tensor Core アーキテクチャには変更が必要になり、GPU ユーザーは今後リリースされる新しい GPU ハードウェアを購入する必要がありそうです。

優れた効率と消費電力

システム レベルで見ると、コンピューティング プラットフォームは、指定された消費電力または熱の範囲内で最大限の処理能力を提供する必要があります。この要件に対応するには、コンピューティング プラットフォームが次の条件を満たしている必要があります。

- 消費電力が許容範囲内に収まる
- 電力バジェットの範囲内で最大限の処理能力を実現する

ザイリンクスは、ユーザーの消費電力と熱の条件に最も適したデバイスを選択できるように、幅広いラインアップの All Programmable デバイスを提供しています。さらに、ザイリンクスの UltraScale+ デバイスは、消費電力を 30% 削減し、効率を 20% 向上させる低電圧モード (V_{LOW}) を備えています。

表 1 に示すように、ザイリンクス デバイスは、シンプルで強力な処理能力という観点から見て、固定された有効ビット数で最も効率的な汎用コンピューティング プラットフォームです。ザイリンクス FPGA ベースのアーキテクチャでは処理に関連するオーバーヘッドが小さくなることが、その主な理由です。たとえば、GPU では、ソフトウェアのプログラマビリティを容易にするために、コンピューティング リソース周囲の複雑性が増しています。現在の深層学習ワークロードのテンソル演算の場合、NVidia 社の Tesla V100 は、強化された Tensor Core によってザイリンクスの FPGA/SoC と同等クラスの効率を提供します。しかし、深層学習ワークロードは急速に進化しています。したがって、NVidia 社の Tensor Core が深層学習ワークロードでどれだけの間高い効率を維持できるかははっきりしません。また、NVidia V100 はその他の汎用ワークロードでの効率が高くないことも明らかです。

表 1: デバイスの効率 (デバイス使用率 90%、アクティブ クロック サイクル 80% の場合)(1)

デバイス	汎用コンピューティングの効率	テンソル演算の効率
NVidia Tesla P4	209GOP/s/W(2) (INT8)	
NVidia Tesla P40	188GOP/s/W (INT8)	
NVidia Tesla V100	72GFLOP/s/W(3) (FP16)	288GFLOP/s/W (FP16)
Intel Stratix 10	136GOP/s/W (INT8)	
ザイリンクス Virtex® UltraScale+™	277GOP/s/W (INT8) [参照 27]	

注記:

1. 引用した数値は比較のみを目的としています。実現可能なデバイス効率は、最終アプリケーションとユーザーによって異なります。
2. 消費電力 1 ワットあたりの Giga Operations Per Second です。
3. 消費電力 1 ワットあたりの Giga Floating Point Operations Per Second です。

このホワイト ペーパーの前半で説明した制限のため、実際のワークロードおよびシステムにおける GPU は表 1 に示した数値の達成に苦戦しています。

ザイリンクス デバイスの柔軟性やその他の利点とザイリンクスの新しいソフトウェア開発スタックを組み合わせることで、ザイリンクス ベースのソリューションは、多様な最終アプリケーションおよびワークロードで GPU よりもはるかに高い効率を実現します。

All Programmable デバイスの柔軟性

ザイリンクス デバイスは、多様な高性能最終システムの処理能力、効率、コスト、柔軟性の要件を満たすように入念に設計されています。ザイリンクスは、TSMC 社の 16nm FinFET プロセス技術などの最先端のプロセス技術で製造されるハードウェア プログラマブル リソース (ロジック、配線、I/O) と柔軟性に優れた統合型コア ブロック (DSP スライスおよび UltraRAM) を組み合わせてこれらの要件をバランスよく満たします。

ハードウェア プログラマビリティと柔軟性に優れたザイリンクス デバイスは、指定されたワークロードの要件に応じて、基盤となるハードウェアをコンフィギュレーションすることが可能です。その後で (実行時でも)、パーシャル リコンフィギュレーションを使用して、データパスを容易に再構築できます [参照 35]。図 4 はザイリンクスの All Programmable デバイスが提供する柔軟性の一部を示したものですが、ザイリンクス デバイスの柔軟性の全容は 1 枚の図では表現できません。カーネル (またはユーザー デザイン要素) は、プログラマブル I/O、その他のカーネル、LUTRAM、ブロック RAM、UltraRAM、外部メモリなどと直接インターフェイスできます。

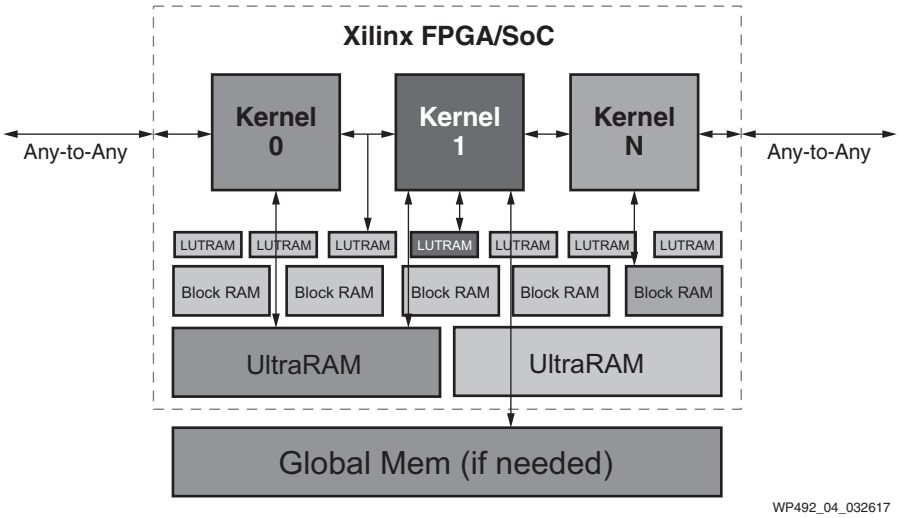

図 4: All Programmable デバイスのデータパスと Any-to-Any I/O

独自のハードウェア プログラマビリティを備えたザイリンクス デバイスは、SIMT や固定されたデータパスなどの制約を受けません。大規模並列処理ワークロード、中規模並列処理ワークロード、パイプライン化されたシーケンシャル ワークロード、それらを組み合わせたワークロードなど、ザイリンクス デバイスの処理能力と効率はあらゆるワークロードに利用できます。さらに、基盤となるアルゴリズムが変更された場合 (機械学習ネットワークの進化など)、プラットフォームはそれに適応できます。

ザイリンクス デバイスの柔軟性は、多くのシステムやワークロードにメリットをもたらします。その例の 1 つが機械学習の推論です。機械学習推論の 1 つのトレンドとして、スパースなネットワークへの移行が挙げられます。ザイリンクス デバイスのユーザーは、既にこれらのトレンドを利用しています。NVidia 社もザイリンクスのユーザーです。音声認識に関する NVidia 社の最近の論文によると、ザイリンクスの FPGA を使用した場合、CPU に比べて実行速度は 43 倍、効率は 40 倍に向上し、NVidia 社の GPU と比べても実行速度は 3 倍、効率は 11.5 倍に向上します [参照 36]。またザイリンクスの FPGA では、プログラマブルなデータパスにより、バッチ処理の必要性が軽減されます。バッチ処理は、システムのレイテンシとリアルタイム パフォーマンスの関係に大きな影響を与えます。

ビッグ データ分析についても、ザイリンクス デバイスの柔軟性には明確な利点があります。ザイリンクスの FPGA は、可変長文字列などの複雑なデータを含む SQL ワークロードについて高い効率と高速性が実証されています。Baidu 社は、ザイリンクスの Kintex® UltraScale™ KU115 デバイス ベースのカードを使用して、分析処理を 25 倍以上高速化しました。このカードの消費電力はわずか 50W であり、Baidu 社のソリューションは GPU インプリメンテーションの 4 倍の効率を示しています [参照 37]。同様にテキストおよびパターン マッチング ワークロードの場合、ザイリンクス ベースの RegEx インプリメンテーションの実行速度は、CPU インプリメンテーションよりも 14.5 ～ 18 倍、GPU インプリメンテーションよりも約 3 倍高速になることを示す研究結果があります [参照 38][参照 39]。

もう 1 つの具体例として、ゲノム分析が挙げられます。ゲノム分析の高速化に GPU を利用した場合、Intel 社の Xeon CPU に比べて 6 ～ 10 倍高速化できることが実証されています [参照 40]。しかし、ザイリンクスの FPGA ではさらに大きな効果が得られ、同等の CPU に比べて 80 倍以上高速化されます [参照 41]。

柔軟性に優れたザイリンクス デバイスは、クラウド サービス プロバイダーの CPaaS (Compute Platform as a Service) の理想的な要素となります。各種の SaaS (Software as a Service) も、ザイリンクス デバイスの利点を享受できます。

さらに、運転支援システムの設計者は、柔軟性に優れたザイリンクス デバイスを使用して、完全な自動運転に至るまでさまざまな SAE レベルに対応するスケーラブルなプラットフォームを構築できます。SAE レベルの詳細は、SAE のウェブサイトを参照してください。ザイリンクス デバイスは、システムのリアルタイム性能とレイテンシの目標値を維持しながら、レーダー、カメラ、超音波センサーなど、さまざまなソースから入力されるセンサー データを効率的に処理できます。

Any-to-Any I/O の柔軟性

デバイスのコンピューティング リソースの柔軟性に加え、ザイリンクスの Any-to-Any I/O の柔軟性は、既存のインフラストラクチャへのシームレスな統合を確実にします。たとえば、これらのデバイスは、ホスト CPU を介さずにネットワークまたはストレージ デバイスに直接接続できます [参照 42]。こうした I/O の柔軟性により、ザイリンクス デバイスは、インフラストラクチャのさまざまな変更や更新に適応できます。

ザイリンクスの UltraScale アーキテクチャ デバイスの詳細は、ザイリンクスのホワイト ペーパー ライブラリを参照してください。

オンチップ メモリ

表 2 に示すように、ザイリンクス デバイスは、柔軟性、高帯域幅、低レイテンシで業界最先端をいく 500MB のオンチップ メモリを備えています [参照 44]。大容量オンチップ メモリ キャッシュにより、ワークロードのメモリ需要の大部分をオンチップ メモリで処理できます。これにより、外部メモリへのアクセスに関連するボトルネックを軽減し、HBM2 などのメモリ帯域幅の大きいソリューションの消費電力とコストを削減できます。たとえば GoogLeNet など、ほとんどの深層学習ネットワーク トポロジの係数/機能マップがオンチップ メモリに格納されるため、コンピューティング効率の向上とコストの削減が可能となります。

表 2: デバイスのオンチップ メモリのサイズ

デバイス	オンチップ メモリ (MB)
NVidia Tesla P4	80
NVidia Tesla P40	101
NVidia Tesla P100	144
NVidia Tesla V100	300
Intel Stratix 10	244
ザイリンクス Virtex® UltraScale+™	500

オンチップ メモリ内で処理が完了するため、オフチップ メモリへのアクセスに関連する大きなレイテンシがなくなり、システムのリアルタイム パフォーマンスを最大化できます。

インパッケージ HBM

追加の高帯域幅メモリが必要な場合のために、ザイリンクスは一部の Virtex UltraScale+ デバイスに HBM を搭載しました。インパッケージ HBM スタックのメモリ帯域幅 460GB/s に加え、ザイリンクス HBM メモリ コントローラーによって柔軟性をさらに高め、デバイスおよび利用可能なメモリ帯域幅にワークロードを効率的に割り当てて、最大限の電力効率と処理効率を引き出します。

機能安全

産業オートメーションや近年の ADAS など、安全性が重視されるアプリケーション要件への対応について、ザイリンクスには長期的な実績があります。ザイリンクスのツールおよびデバイスは、機能安全アプリケーションのサポートを意図してゼロから設計され、適切なレベルの認証を取得しています [参照 45]。

その結果、Zynq®-7000 All Programmable SoC は、多くの自動車メーカーで安全性が重視される ADAS アプリケーションに採用されています。Zynq UltraScale+ MPSoC は、機能安全が重視されるアプリケーションをさらに幅広くサポートします。

最後に

システム設計者は、コンピューティングの新しい時代における難しい選択に直面しています。ザイリンクスの FPGA/SoC は、システム設計者が将来システムの基本的要件と課題に対応する際に発生するリスクを最小限に抑え、高い柔軟性で将来にわたるプラットフォームの有効性を確保します。

深層学習については、UltraScale アーキテクチャ内の DSP アーキテクチャ特有の並列性が、INT8 ベクトル内積のスケーラブルなパフォーマンスを発揮し、ニューラル ネットワークのたたみ込みおよび行列乗算のスループットを向上させます。この効果は、深層学習の推論のレイテンシを削減します。高速 DSP アレイと極めて効率的なブロック RAM メモリ階層を組み合わせ、さらに UltraRAM メモリ アレイを加えてクラス最高の電力効率を実現します。

ザイリンクス デバイスの利点は、https://japan.xilinx.com/products/boards-and-kits.html に示す開発キットと、HLS、SDSoC、SDAccel など各種デザイン入力ツールを利用することで確認できます。

参考資料

1. ZDNET『Vision and neural nets drive demand for more powerful chips』アクセス日: 2017 年 4 月 6 日
http://www.zdnet.com/article/vision-and-neural-nets-drive-demand-for-better-chips/
2. NVidia Corporation
http://www.nvidia.com/object/IO_20020111_5424.html
3. MJ Misic、DM Durdevic、MV Tomasevic『Evolution and trends in GPU computing』MIPRO、2012 Proceedings of the 35th International Convention、289-294
http://ieeexplore.ieee.org/abstract/document/6240658/
4. Nicole Hemsoth『FPGAs Focal Point for Efficient Neural Network Inference』最終アクセス日: 2017 年 4 月 6 日
https://www.nextplatform.com/2017/01/26/fpgas-focal-point-efficient-neural-network-inference/
5. http://www.cse.psu.edu/hpcl/docs/2016_PACT_Onur.pdf
6. Babak Falsafi、Bill Dally、Desh Singh、Derek Chiou、Joshua J. Yi、Resit Sendag 『FPGAs versus GPUs in Datacenters』IEEE Micro、vol. 37、no. 1、pp. 60-72、2017 年 1 月
http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7866802
7. Richard Vuduc、Aparna Chandramowlishwaran、Jee Choi、Murat Guney、Aashay Shringarpure『On the limits of GPU acceleration』HotPar'10 Proceedings of the 2nd USENIX conference on Hot topics in parallelism、13-13 ページ。Berkeley, CA 2010 年 6 月 14 ～ 15 日
http://hpcgarage.org/wp/vuduc2010-hotpar-cpu-v-gpu.pdf
8. Fowers, J.、Ovtcharov, K.、Strauss, K.、Chung, E.S.、Stitt, G.『A High Memory Bandwidth FPGA Accelerator for Sparse Matrix-Vector Multiplication』IEEE Int. Symp. on Field-Programmable Custom Computing Machines (2014)
http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6861585
9. B. Liu、M. Wang、H. Foroosh、M. Tappen、M. Pensky『Sparse Convolutional Neural Networks』Computer Vision and Pattern Recognition (CVPR)、2015 IEEE Conference
http://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Liu_Sparse_Convolutional_Neural_2015_CVPR_paper.pdf
10. Yaman Umuroglu ほか『Random Access Schemes for Efficient FPGA SpMV Acceleration』Microprocessors & Microsystems Volume 47 Issue PB、2016 年 11 月、321-332 ページ
http://www.sciencedirect.com/science/article/pii/S0141933116300023 (図 4 - 使用率を示す)
11. Sparsh Mittal、Jeffrey S. Vetter『A Survey of CPU-GPU Heterogeneous Computing Techniques』ACM Computing Surveys (CSUR) Volume 47 Issue 4、2015 年 7 月、Article No. 69
12. ザイリンクス ホワイト ペーパー『浮動小数点から固定小数点への変換による消費電力およびコストの削減』最終アクセス日: 2017 年 4 月 6 日
https://japan.xilinx.com/support/documentation/white_papers/wp491-floating-to-fixed-point.pdf
13. Marc Baboulin ほか『Accelerating Scientific Computations with Mixed Precision Algorithms』Computer Physics Communications 180 (2009 年) 2526-2533
http://www.netlib.org/utk/people/JackDongarra/PAPERS/202_2009_Accelerating-Scientific-Computations-with-Mixed-Precision-Algorithms.pdf
14. Suyog Gupta ほか『Deep Learning with Limited Numerical Precision』ICML'15 Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37、1737-1746 ページ
https://arxiv.org/pdf/1502.02551.pdf

15. William Dally『High-Performance Hardware for Machine Learning』最終アクセス日: 2017 年 4 月 6 日
https://media.nips.cc/Conferences/2015/tutorialslides/Dally-NIPS-Tutorial-2015.pdf
16. NVidia Corporation『NVIDIA TensorRT』最終アクセス日: 2017 年 4 月 6 日
https://developer.nvidia.com/tensorrt
17. Xinxin Mei、Xiaowen Chu『Dissecting GPU Memory Hierarchy through Microbenchmarking』IEEE Transactions on Parallel and Distributed Systems Volume: 28、Issue: 1、72-86 ページ、2017 年 1 月 1 日
https://arxiv.org/pdf/1509.02308.pdf
18. NVidia Corporation『Cuda C Programming Guide』最終アクセス日: 2017 年 4 月 6 日
https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html#memory-hierarchy
19. Mark Gebhart ほか『Unifying Primary Cache, Scratch, and Register File Memories in a Throughput Processor』MICRO-45 Proceedings of the 2012 45th Annual IEEE/ACM International Symposium on Microarchitecture、96-106 ページ、Vancouver, B.C.、CANADA - 2012 年 12 月 1 ～ 5 日
https://research.nvidia.com/sites/default/files/publications/Gebhart_MICRO_2012.pdf
20. Chris Edwards『Minimize Memory Moves for Greener Data Centers』最終アクセス日: 2017 年 4 月 6 日
http://www.techdesignforums.com/blog/2016/06/08/dac-data-center-acclerators/
21. Vincent Vanhouckehttps ほか『Improving the Speed of Neural Networks on CPUs』Proc.Deep Learning and Unsupervised Feature Learning NIPS Workshop 2011 [オンライン] 利用可能:
http://research.google.com/pubs/archive/37631.pdf
22. Christian Szegedy ほか『Going Deeper with Convolutions』Proceedings Computer Vision and Pattern Recognition (CVPR)、1–9 ページ、2015 年
http://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Szegedy_Going_Deeper_With_2015_CVPR_paper.pdf
23. Funding Universe『Xilinx, Inc. History』最終アクセス日: 2017 年 4 月 6 日
http://www.fundinguniverse.com/company-histories/xilinx-inc-history/
24. Xilinx Inc.『ソフトウェア ゾーン』最終アクセス日: 2017 年 4 月 6 日
https://japan.xilinx.com/products/design-tools/software-zone.html
25. Xilinx Inc.『アクセラレーション ゾーン』最終アクセス日: 2017 年 4 月 6 日
https://japan.xilinx.com/products/design-tools/acceleration-zone.html#libraries
26. ANANDTECH『NVIDIA Announces Tesla P40 & Tesla P4 - Neural Network Inference』最終アクセス日: 2017 年 4 月 6 日
27. ザイリンクス ホワイト ペーパー『ザイリンクス デバイスでの INT8 に最適化した深層学習の実装』
最終アクセス日: 2017 年 4 月 6 日
https://japan.xilinx.com/support/documentation/white_papers/wp486-deep-learning-int8.pdf
28. Philipp Gysel ほか『Hardware-Oriented Approximation of Convolutional Neural networks』ICLR 2016
https://arxiv.org/pdf/1604.03168v3.pdf
29. Chenzhuo Zhu ほか『Trained ternary quantization』ICLR 2017
https://arxiv.org/pdf/1612.01064.pdf
30. Yaman Umuroglu ほか『FINN: A Framework for Fast, Scalable Binarized Neural Network Inference』25th International Symposium on FPGAs、2017 年 2 月
https://arxiv.org/pdf/1612.07119.pdf
31. Wonyong Sunget ほか『Resiliency of Deep Neural Networks under Quantization』ICLR 2016
https://arxiv.org/pdf/1511.06488v3.pdf
32. Nicholas J. Fraser ほか『Scaling Binarized Neural Networks on Reconfigurable Logic』HiPEAC 2017
https://arxiv.org/abs/1701.03400
33. Cl´ement Farabet、Yann LeCun、Koray Kavukcuoglu、Eugenio Culurciello、Berin Martini、Polina Akselrod、Selcuk Talay『Large-Scale FPGA-based Convolutional Networks』
http://yann.lecun.com/exdb/publis/pdf/farabet-suml-11.pdf
34. C. Zhang、P. Li、G. Sun、Y. Guan、B. Xiao、J. Cong『Optimizing FPGA-based Accelerator Design for Deep Convolutional Neural Networks』ACM/SIGDA ISFPGA、161–170 ページ、ACM、2015 年
https://pdfs.semanticscholar.org/2ffc/74bec88d8762a613256589891ff323123e99.pdf
35. Xilinx, Inc.『Vivado Design Suite でのパーシャル リコンフィギュレーション』最終アクセス日: 2017 年 4 月 6 日
https://japan.xilinx.com/products/design-tools/vivado/implementation/partial-reconfiguration.html
36. Song Han ほか『ESE: Efficient Speech Recognition Engine with Sparse LSTM on FPGA』International Symposium on FPGA 2017
https://arxiv.org/pdf/1612.00694.pdf

37. Jian Ouyang ほか『SDA: Software-Defined Accelerator for General-Purpose Big Data Analysis System』Hotchip 2014
http://www.hotchips.org/wp-content/uploads/hc_archives/hc28/HC28.23-Tuesday-Epub/HC28.23.80-Big-Data-Epub/HC28.23.832-SDA-BD-Analysis-JianOuyang-Baidu-v06.pdf

38. Shreyas G Singapura ほか『FPGA Based Accelerator for Pattern Matching in YARA Framework』CENG 2015
http://ceng.usc.edu/techreports/2015/Prasanna%20CENG-2015-05.pdf

39. Yuichiro Utan ほか『A GPGPU Implementation of Approximate String Matching with Regular Expression Operators and Comparison with Its FPGA Implementation』PDPTA 2012
https://pdfs.semanticscholar.org/2667/ac95d36ab63ae6eeb4b352f4c20dc46344c3.pdf

40. BarraCUDA『The BarraCUDA Project』最終アクセス日: 2017 年 4 月 6 日
http://seqbarracuda.sourceforge.net/index.html

41. Edico Genome『DRAGEN Genome Pipeline』最終アクセス日: 2017 年 4 月 6 日
http://www.edicogenome.com/wp-content/uploads/2015/02/DRAGEN-Genome-Pipeline.pdf

42. Microsoft 社の論文 『A Cloud-Scale Acceleration Architecture』最終アクセス日: 2017 年 4 月 6 日
https://www.microsoft.com/en-us/research/wp-content/uploads/2016/10/Cloud-Scale-Acceleration-Architecture.pdf

43. Xilinx, Inc. 『UltraScale アーキテクチャ』最終アクセス日: 2017 年 4 月 6 日
https://japan.xilinx.com/support/documentation/white_papers/wp470-ultrascale-plus-power-flexibility.pdf

44. ザイリンクス ホワイト ペーパー『UltraRAM: UltraScale+ デバイスに搭載された画期的なエンベデッド メモリ』
最終アクセス日: 2017 年 4 月 6 日
https://japan.xilinx.com/support/documentation/white_papers/wp477-ultraram.pdf

45. ザイリンクス ホワイト ペーパー『IEC61508 および ISO26262 に準拠した安全アプリケーションのリスク軽減と効率向上』
最終アクセス日: 2017 年 4 月 6 日
https://japan.xilinx.com/support/documentation/white_papers/wp461-functional-safety.pdf

改訂履歴

次の表に、この文書の改訂履歴を示します。

日付	バージョン	内容
2017 年 6 月 13 日	1.0.1	誤字の訂正。
2017 年 6 月 8 日	1.0	初版

免責条項

本通知に基づいて貴殿または貴社 (本通知の被通知者が個人の場合には「貴殿」、法人その他の団体の場合には「貴社」。以下同じ) に開示される情報 (以下「本情報」といいます) は、ザイリンクスの製品を選択および使用することのためにのみ提供されます。適用される法律が許容する最大限の範囲で、(1) 本情報は「現状有姿」、およびすべて受領者の責任で (with all faults) という状態で提供され、ザイリンクスは、本通知をもって、明示、黙示、法定を問わず (商品性、非侵害、特定目的適合性の保証を含みますがこれらに限られません)、すべての保証および条件を負わない (否認する) ものとします。また、(2) ザイリンクスは、本情報 (貴殿または貴社による本情報の使用を含む) に関係し、起因し、関連する、いかなる種類・性質の損失または損害についても、責任を負わない (契約上、不法行為上 (過失の場合を含む)、その他のいかなる責任の法理によるかを問わない) ものとし、当該損失または損害には、直接、間接、特別、付随的、結果的な損失または損害 (第三者が起こした行為の結果被った、データ、利益、業務上の信用の損失、その他あらゆる種類の損失や損害を含みます) が含まれるものとし、それは、たとえ当該損害や損失が合理的に予見可能であったり、ザイリンクスがそれらの可能性について助言を受けていた場合であったとしても同様です。ザイリンクスは、本情報に含まれるいかなる誤りも訂正する義務を負わず、本情報または製品仕様のアップデートを貴殿または貴社に知らせる義務も負いません。事前の書面による同意のない限り、貴殿または貴社は本情報を再生産、変更、頒布、または公に展示してはなりません。一定の製品は、ザイリンクスの限定的保証の諸条件に従うこととなるので、https://japan.xilinx.com/legal.htm#tos で見られるザイリンクスの販売条件を参照してください。IP コアは、ザイリンクスが貴殿または貴社に付与したライセンスに含まれる保証と補助的条件に従うことになります。ザイリンクスの製品は、フェイルセーフとして、または、フェイルセーフの動作を要求するアプリケーションに使用するために、設計されたり意図されたりしていません。そのような重大なアプリケーションにザイリンクスの製品を使用する場合のリスクと責任は、貴殿または貴社が単独で負うものです。https://japan.xilinx.com/legal.htm#tos で見られるザイリンクスの販売条件を参照してください。

自動車用のアプリケーションの免責条項

オートモーティブ製品 (製品番号に「XA」が含まれる) は、ISO 26262 自動車用機能安全規格に従った安全コンセプトまたは余剰性の機能 (「セーフティ設計」) がない限り、エアバッグの展開における使用または車両の制御に影響するアプリケーション (「セーフティ アプリケーション」) における使用は保証されていません。顧客は、製品を組み込むすべてのシステムについて、その使用前または提供前に安全を目的として十分なテストを行うものとします。セーフティ設計なしにセーフティ アプリケーションで製品を使用するリスクはすべて顧客が負い、製品の責任の制限を規定する適用法令および規則にのみ従うものとします。

この資料に関するフィードバックおよびリンクなどの問題につきましては、jpn_trans_feedback@xilinx.com まで、または各ページの右下にある [フィードバック送信] ボタンをクリックすると表示されるフォームからお知らせください。いただきましたご意見を参考に早急に対応させていただきます。なお、このメール アドレスへのお問い合わせは受け付けておりません。あらかじめご了承ください。