FPGA ベースのボリュームレイキャスティング

はじめに：

このドキュメントには、Super Computing 2017 で行った Alpha Data 社の FPGA ベースの Volume Ray-Casting のデモについての説明が含まれています。このデモでは、Vivado HLS と Alpha Data の ADB3 PCIe Bridge を利用して、PCI Express インタフェースを備えた FPGA カードに計算集中型のアルゴリズムを実装する方法を示します。使用したシステムは、スケーラブルかつ電力効率的であることが示されており、FPGA プラットフォームは、トップレベルの IP インテグレータ（IPI）システム、オン/オフチップデータフロー用 IP コア、ボード固有のホスト通信の組み合わせで構築されています。これらは、C ++で記述され、ザイリンクスの Vivado HLS ツールで合成されたレイキャスティング IP コアと組み合わされています。これらの IP の重要な側面について説明します。

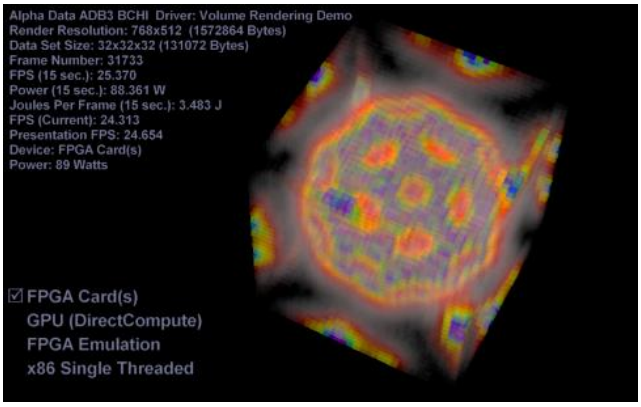

Figure 1 ADM-PCIE-KU3 で動作するデモ

このアルゴリズムには次の 3 つのフェーズがあります。

1）データをアップロードしてタスクを分割する
2）レンダリングアルゴリズムを実行する
3）結果をダウンロードして再組み立てする

実行段階では、出力の各ピクセルは他のピクセルとは無関係に計算できます。 これにより、タスクを分割（分解）し、ピースを複数の実行ユニットに分散させることで、スケーラビリティが簡単になります。
このデモンストレーション FPGA デザインの構造は、タスクを独立した作業項目に分解することを可能にする他の計算集中型アルゴリズムにも適しています。出力例を Figure 1 に示します。

ここでは、1 秒あたりの反復数に関して全体的に達成されたパフォーマンスが、使用可能な FPGA カードの数に応じてどのように変化するかを説明します。結果は、CPU に代わるものと比較して FPGA 実装のエネルギー効率が向上していることを示しています。ノード内の FPGA と CPU の比率を上げてスケーリングすると、ノードレベルでのエネルギー効率がさらに向上します。現在のバージョンのデモでは、使用可能な FPGA カード数の制限はホストシステムの PCI Express スロットの数によって決まりますが、原則としてこのタスクは異種の相互接続（イーサネットなど）を持つ FPGA カードはいくつにでも分解できます。ただし、いくつかのボトルネックがシステム全体のスケーラビリティに実用上の限界を課すことが予想され、これは将来の研究で調査されるでしょう。

株式会社ミッシュインターナショナル 〒190-0004 東京都立川市柏町 4-56-1
TEL: 042-538-7650 https://www.mish.co.jp E-mail: sales@mish.co.jp

アーキテクチャ：

Volume Ray-Casting の実行は 3 つのフェーズに分けられます。

1）体積測定データセットを装置にアップロードし、レンダリングパラメータを設定する
2）Vivado HLS で生成された IP を「C to gate」IP を使用してレンダリングアルゴリズムを実行する
3）デバイスから結果（レンダリングされた画像）をダウンロードする

学術的な観点から、フェーズ 1）と 3）は特に興味深いものではなく、目新しいものではありません。ただし、フェーズ 2）はこのドキュメントの焦点となります。レンダリング用のパイプラインは一連のコンポーネントで構成され、各コンポーネントは C ++で記述され、Vivado HLS ツールを使用してネットリストに合成されます。これらのコンポーネントは、Vivado のブロックダイアグラムエディタが提供する配線図を使用して、ADB3 PCIe ブリッジ（BCHI：Board Control and Host Interface）のインスタンスと共に接続され、Figure 2 に示すような完全なレンダリングアーキテクチャを作成します。

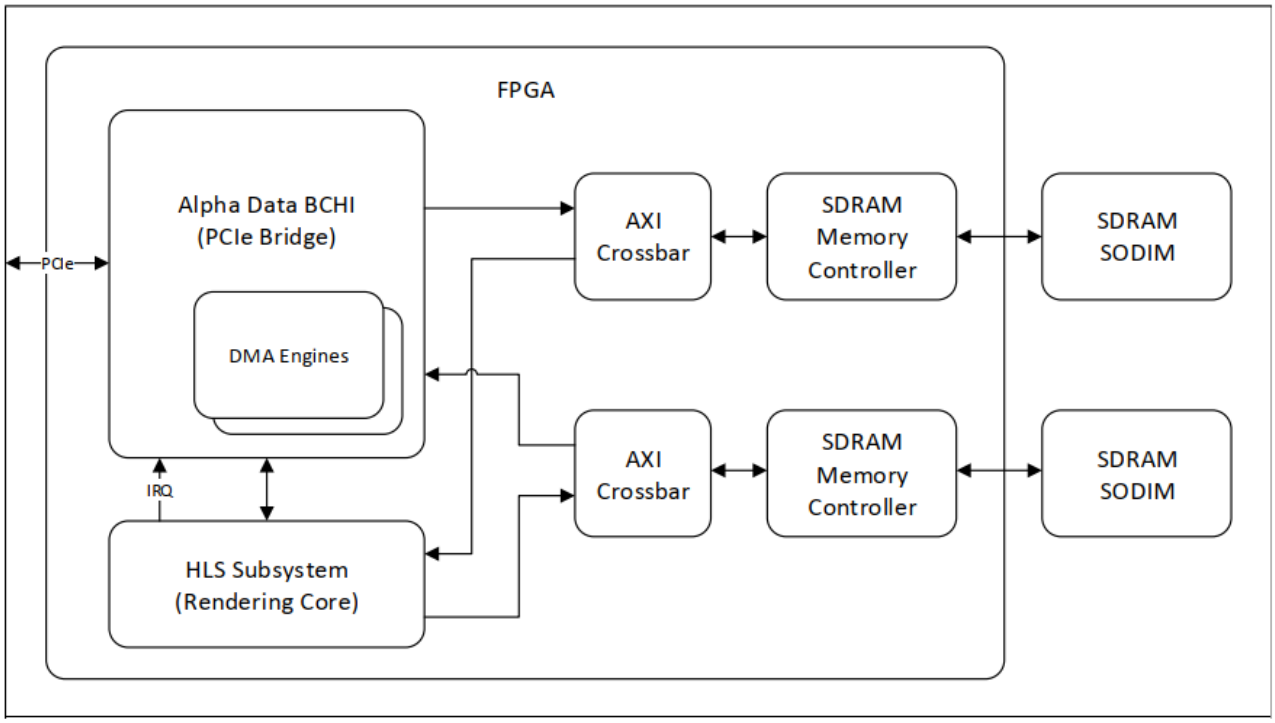

Figure 2　FPGA デザインのトップレベル概略ブロック

このデモで選択された FPGA カードは ADM-PCIE-KU3 です。これには、FPGA 用のオフチップメモリを提供する 2 つの SODIMM スロットがあります。各 SODIMM にはメモリコントローラが必要です。クロスバーは、（a）ホストシステム（データと結果のアップロードとダウンロードのため）、および（b）レンダリングアルゴリズムを実行する HLS サブシステムの両方

Figure 3　Alpha Data 製　ADM-PCIE-KU3

株式会社ミッシュインターナショナル　〒190-0004 東京都立川市柏町 4-56-1
TEL: 042-538-7650　https://www.mish.co.jp　E-mail: sales@mish.co.jp

によって、メモリへの共有アクセスを可能にします。
各 FPGA カードで、ホストインタフェースは BCHI によって提供されます。この IP は最大 4 つの DMA エンジンを含む PCI Express エンドポイントを提供し、CPU が HLS サブシステムを制御する FPGA 内のレジスタを読み書きすることも可能にします。ホストを中断する機能も BCHI によって提供されます。これは、作業項目の完了をホストに通知するために HLS サブシステムによって使用されます。

レンダリングアルゴリズム：

このデモで使用されている Volume Ray-Casting アルゴリズムは次のとおりです。
レンダリングされる各フレームについて、出力画像の各ピクセルは、視点からのピクセルの位置（スクリーン上）を通る光線として投影されます。光線がピクセルの位置から投影されると、光線がデータセットのボリュームと交差するかどうかを判断するために、最初のカリング計算が実行されます（データセットで表される 3 次元空間を含む直方体）。結果が負の場合、出力ピクセルは背景色に設定され、それ以上の処理は不要です。体積データセット（ボクセルの３次元配列）は、体積を３次元ボクセルの集合と見なすのとは対照的に、２次元テクスチャ平面の３つのスタック、すなわち XY、XZ および YZ として表すことができます。この表現は、アルゴリズムがどのように機能するかの鍵となります。最初のカリングテスト（上記参照）をパスした光線は、3 つの平面スタックに対してキャストされ、各平面スタックの最初のチェックで最もスクリーンに近い衝突が返されるように順序付けされます（最短の光線になります）。各スタックの最初の衝突がボリュームの境界内で検出されるまで、各スタックの衝突が繰り返されます（プレーンのサイズは無限大です）。各平面スタックについての衝突の結果を比較すると、最も早いものが次の交差点として選択される。特定の２次元平面とその２次元平面内の点が決定されると、その点はボリュームデータセット内の３次元ボクセルの座標に変換されます。次に、体積測定データを含むメモリにアクセスしてボクセルのカラー値を読み取ります。このカラー値は現在のレイのカラーとブレンドされます。ボクセルが透明な色の場合は、光線は通過します。不透明なボクセルに当たる光線、またはボリュームのより深い浸透が見えなくなるようなアルファ値になる光線は、それらが遭遇する最後の衝突を反映するように色の値を更新した後にフレームバッファに書き出されます。
３組の平面に対して終了条件：（ｉ）不透明な衝突のために旅は終了する（ｉｉ）さらなる衝突は目に見える結果をもたらさない（iii）光線がボリュームから完全に出る、　に達するまで、平面の組のより深部まで続く光線はさらに処理されます。
この状態に達すると、それ以上光線を処理する必要はなくなり、その色の値がフレームバッファに書き出されます。すべてのピクセルが計算された（すなわち、すべての光線が投射され終了した）とき、フレームはダウンロードおよび提示（表示）の準備ができたことになります。

レンダリング実装：

上記のレンダリングアルゴリズムの説明は、どの実装も算術集約的であり、メモリへのランダムアクセスを必要とすることを示唆しています。HDL ですべてをコーディングするよりも、Vivado HLS および他のいわゆる「C to Gates」ツールが提供する主な利点は、そのようなアルゴリズムを高級言語でコーディングできることです。

株式会社ミッシュインターナショナル　〒190-0004 東京都立川市柏町 4-56-1
TEL: 042-538-7650　https://www.mish.co.jp　E-mail: sales@mish.co.jp

Vivado HLS を正しく使用することで非常に強力なツールになります。他の EDA ツールやプログラミング言語と同様に、理解しなければならない制限事項もあります。Vivado HLS の主な強みは、複雑な算術機能を実行するために IP ブロックを迅速に開発できることです。VHDL や Verilog などの従来の HDL 言語でコーディングするには長い時間がかかります。一度記述すれば、HLS コードは簡単に変更できますが、設計への構造的な変更を必要とする HDL コードへの変更には膨大な開発期間が必要になる場合があります。たとえば、C/C++では、ソフトウェアエンジニアは複雑な算術式を評価する関数を書くのに数分かかることがあります。従来の HDL 言語で同じ式をコーディングするためには、エンジニアはスペースだけでなく時間も考慮しなければならず、パイプライン化やタイミングクロージャを達成する必要性などのパフォーマンスも考慮する必要があります。

また、Vivado HLS にはいくつかの制限があります。シーケンシャルプロセッサやプロセッサの配列（GPU コアなど）用に書かれたコードを魔法のように分析することはできず、そのコードで記述されたアルゴリズムを実行するための最適なハードウェアアーキテクチャを作成することはできません。何もせずに単純にアルゴリズムを HLS コードに変換する HLS の制限を考慮すると正しい実装が作成されますが、パフォーマンスが低下します。アルゴリズムに適したアーキテクチャを作成するには、アルゴリズムをパイプライン形成できるサブコンポーネントに分解するために、何らかの分析を実行する必要があります。

次に考慮すべき点は FPGA アーキテクチャです。アルゴリズムが FPGA 内のプリミティブにどのようにマッピングされる可能性があるかについての知識は、FPGA リソースを不必要に消費するのを避けるために役立ちます。

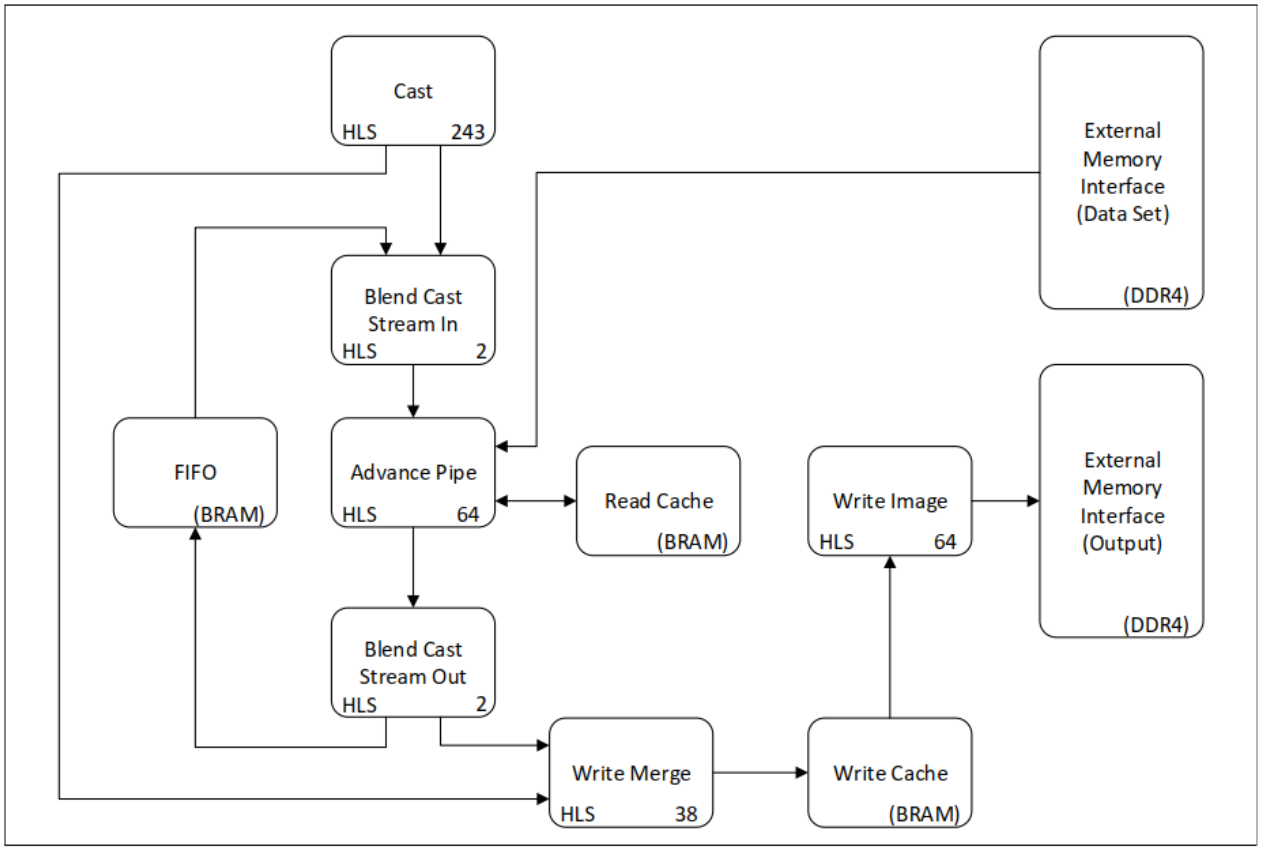

Figure 4 HLS サブシステムの概略ブロック（レンダリングコア）

Figure 4 は、HLS 生成コンポーネントと非 HLS コンポーネントの接続性を示しています。各 HLS コンポーネントには、そのパイプラインの深さが記されています。Cast コンポーネントは、ボリュームデータにキャストされる初期光線のストリームを生成します。画面上の各ピクセル（作業項目）に対して、他のコンポーネントで使用されている光線を投影して初期化するために 3 次元幾何学計算を実行します。処理された各作業項目は、2 つの可能な出力ストリーム（ｉ）ボリュームデータを見逃すように計算された光線のためのバイパスストリーム、（ｉｉ）ボリュームデータと交差し処理を必要とする光線のためのストリーム。のうちの 1 つに出力されます。

Cast コンポーネントは、数学的には最も重い HLS 生成コンポーネントです。HLS はこのコンポーネント用に 243 ステージのパイプラインを生成し、出力ストリームがブロックされていない限り、クロックサイクルごとに新しい作業項目を開始できます。作業項目が処理される割合は、繰り返し間隔（II）と呼ばれることがよくあります。キャストのレイテンシは 243、II は 1 です。

HLS コンポーネントのいくつかは非常に単純なストリームマージエンティティです： Blend Cast Stream Out 及び Blend Cast Stream In。HDL で記述するのではなく HLS IP として作成すると、他の HLS IP と互換性があることが保証されているインタフェースがあり、機能検証中に FPGA デザインをマイクロプロセッサでエミュレートするときに使用できます。Blend Cast Stream In と Blend Cast Stream Out の両方の II は 1 です。Advance Pipe は光線が通過するボクセルに基づいて光線の色を変更するためにデータセットを介して光線を進めます。必要な計算のいくつかは、Cast コンポーネントですでに実行されたものと共通しています。したがって、これらの結果を再計算する代わりに、Cast コンポーネントからの結果が光線と共にデータとして渡されます。これにより、アドバンスパイプの複雑さが軽減されるため、パイプラインの長さは 64 だけで済みます。Cast コンポーネントからのストリームに余分なバス幅が追加されると、追加の FPGA 配線リソースが消費されるため、これは無償ではありません。よって、極端に広いバスはタイミングクロージャを達成するのを困難にすることがあります。

アドバンスパイプの II は 1 ですが、外部メモリへのアクセスが原因で、外部メモリがすばやく応答しない場合は停止します。外部メモリへのアクセス数を減らしてストールの可能性を減らすために、BlockRAM ベースのキャッシュがコンポーネントに接続されています。キャッシュの制御は HLS コードで実装されていますが、その実装を完全に制御するためにキャッシュは HDL でコーディングされています。

Advance Pipe コンポーネントを通過するたびに、光線の色に対する 1 つの更新が計算されます。光線がさらに処理を必要とする場合、光線は Blend Cast Stream Out コンポーネントによって FIFO コンポーネントにプッシュされ、後で Advance Pipe に再び入ります。光線がデータセットをたどってその行程を完了した場合は、Write Merge コンポーネントにプッシュされます。Write Merge には 2 つの仕事があります。 キャストバイパスストリームまたは完成した Advanced Pipe 作業アイテムのストリームから完成した作業アイテムを取り出し、2 つのストリームをマージして、そのアイテムをキャッシュに書き込みます。 このコンポーネントの II は 1 です。

Write Image コンポーネントは、ラインの準備が整ったときに Write Cache からキャッシュラインを書き込みます。このコンポーネントの II は 2 です。メモリの幅が 512 ビットであるのに対し、Write Image に入力される項目の幅は 32 ビットなので、ボトルネックにはなりません。したがって、Write Merge は、システムにボトルネックを発生させることなく、最大 16 の II で実行できます。

このセクションでは、Volume Ray-Casting に必要なすべての HLS コンポーネントの概要を説明しました。 これらはそれぞれ II が 1 であるため（上記のように、Write Image を除く）、最大のスループットが達成されます。

ホストインタフェース：

FPGA、Vivado、Vivado-HLS は、単一アルゴリズムの実行専用のカスタムプロセッサを構築するための効果的なツールです。しかし、実行ユニットを持つことは、そのアルゴリズムを使用するためのアプリケーションを作成するための解決策の半分に過ぎません。 問題の他の部分はどのようにしてデータを実行ユニットに出し入れするか、そしてそれを制御するかです。

Alpha Data の ADB3 Bridge and Driver は、この例で示されているカスタム FPGA アクセラレータデバイスとのやり取りという共通の作業を効率化するための多数の機能を提供します。 デバイスへのダイレクトスレーブアクセス、ダイレクトメモリアクセス（DMA）（ホストと FPGA デバイス間の双方向）、割り込み処理、および共通 API を介した複数の ADB3 デバイスとの通信により、デバイス間でタスクを分割できます。

ホストソフトウェア：

アプリケーションのプレゼンテーション層は Direct3D（Microsoft DirectX API の一部）を使用しています。これは、FPGA からレンダリングされたフレームを最小の待ち時間で画面に表示するために使用されています。FPGA カードからレンダリングされたフレームの提示に加えて、直接計算およびシングルスレッド x86_64 バージョンのレンダリングアルゴリズムも実行することができます。

FPGA カードとの相互作用は、フレームの取得とそれらを表示することと同期させるレンダリングループにおいて実行されます。レンダリングループが繰り返されるたびに、レンダリングされたフレームが FPGA カードから収集され、ビューパラメータが更新され、次のフレームのレンダリングが開始されます。

FPGA バージョンのレンダリングルーチンを実行する際のデモアプリケーションは、システムに存在するすべての FPGA デバイスを列挙します。レンダリングタスクは、利用可能なデバイス間で分割されます。 このアプリケーションの場合、利用可能なカード間で作業項目を均等に分割するだけです。

たとえば、システムに 2 枚のカードが存在する場合、フレームは 2 つのハーフフレームに分割され、各 FPGA カードはこれらのハーフフレームの 1 つをレンダリングします。その後、FPGA カードからホストメモリへの DMA 転送の一部としてハーフフレームが結合されてフルフレームになります。

実行と評価：

Volume Ray-Casting の例では、さまざまなパラメータを変更できます。以下に示す結果は、FPGA カードが従来のマイクロプロセッサベースの実装よりも優れている一方で、エネルギー効率が高いことを示しています。以下に詳述する結果を記録するために使用されたホストシステムは、MSI Z87-G45 コンシューマクラスマザーボード上の i7-4770S 3.1GHz CPU と 16 GB の DDR4 SDRAM で構成されています。電力測定は、システムに電力を供給する Corsair AX860i 電源に組み込まれたセンサーを使用して行われました。各実験では、未使用のアクセラレータデバイスをシステムから削除したため、未使用のデバイスがアイドル状態で静的電力を消費することなく公平な比較を行うことができました。

株式会社ミッシュインターナショナル 〒190-0004 東京都立川市柏町 4-56-1
TEL: 042-538-7650 https://www.mish.co.jp E-mail: sales@mish.co.jp

以下の結果を得るために使用された体積測定データは、Figure 1 のような半透明バッキーボールの３２×３２×３２ボクセルセットである。レンダリングを 15 秒間連続して行い、平均フレームレート、電力、およびエネルギー消費量を記録しました。一連のＦＰＧＡカード上でのレンダリングに加えて、適切に移植されたアルゴリズムのバージョンを使用して他の２つのコンピューティングデバイス上でレンダリングも行われた。Microsoft DirectCompute 下で、（ｉ）システムのプロセッサ上の単一スレッド、および（ｉｉ）プロセッサ統合 GPU。

結果：

Table 1 は、さまざまなサイズの出力画像でのシステムの平均達成フレームレートを示しています。（数値が高いほど良い）。 全ての場合において、FPGA カードは CPU より性能が優れていることが分かります。

Frame Size	KU3	2xKU3	3xKU3	x86_64	DXCompute
128x128	228.3	425.02	526	32.96	93.23
256x256	60.65	116.8	143.5	8.27	24.17
512x512	15.35	30.04	37.55	2.131	6.226
1024x1024	3.861	7.624	9.5	0.535	1.587

Table 1　レンダリングスピード（Frame per second）

Table 2 は各実験における平均出力を示しています。これはシステムの平均総電力です（出力画像の表示に使用される電力を含む）。x86_64 シングルスレッドバージョンには最低の消費電力ですが、フレームレートが最も低いことを考えれば、これは驚くべきことではありません。

Frame Size	KU3	2xKU3	3xKU3	x86_64	DXCompute
128x128	101.9	149.5	184.5	76.63	117.4
256x256	89.97	129.6	158.3	78.08	112.5
512x512	87.71	117	141.7	81.39	114.2
1024x1024	84.72	116.4	143	76.11	112

Table 2　システム消費電力（Watt）

表 3 と表 4 に、各実験の各フレームと各ピクセルのエネルギーコストを示します。

Frame Size	KU3	2xKU3	3xKU3	x86_64	DXCompute
128x128	0.446	0.352	0.351	2.325	1.26
256x256	1.483	1.109	1.104	9.441	4.656
512x512	5.716	3.893	3.774	38.2	18.34
1024x1024	21.95	15.27	15.04	142.2	70.6

Table 3　フレームエネルギー（Joule）

Frame Size	KU3	2xKU3	3xKU3	x86_64	DXCompute
128x128	27.22	21.48	21.42	141.91	76.90
256x256	22.63	16.92	16.85	144.06	71.04
512x512	21.80	14.85	14.40	145.72	69.96
1024x1024	20.93	14.56	14.34	135.61	67.33

Table 4　ピクセルエネルギー（μJoule）

株式会社ミッシュインターナショナル　〒190-0004 東京都立川市柏町 4-56-1
TEL: 042-538-7650　https://www.mish.co.jp　E-mail: sales@mish.co.jp

これらの結果は、FPGA カードが、このアプリケーションで x86_64 CPU よりもエネルギー効率が良く高速であることを示しています。

まとめ：

このデモでは、高度なツールを使用して計算集約型のアルゴリズムを FPGA ベースのハードウェアに実装し、複数の FPGA で実行できるように拡張できることを示し、標準の x86 64 CPU と比べてエネルギー効率の高いソリューションを提供できることを検証しました。

Alpha Data 社について

Alpha Data は、1993 年に設立され、計算集約型アプリケーションをターゲットとした最先端の FPGA ソリューションを提供しており、FPGA アクセラレータのマーケットリーダとして市場を牽引しています。主な製品は VPX、XMC、PMC、PCI、CompactPCI、PCIExpress、VXS、VME などの A/D, D/A, FPGA ボードや CameraLink 等のデジタル I/F を搭載した信号処理ボードです。ボーイング、ロックウェルコリンズ、JPL、ロッキードマーチン、モトローラ、BAE などのミリタリ向けやデータセンター向けに広く採用されています。Alpha Data 社の詳細については、https://www.alpha-data.com/を参照してください。

株式会社ミッシュインターナショナル 〒190-0004 東京都立川市柏町 4-56-1
TEL: 042-538-7650 https://www.mish.co.jp E-mail: sales@mish.co.jp