



ホワイトペーパー

NVIDIA Tegra 4 ファミリの CPU アーキテクチャ

4-PLUS-1 クアッドコア

目次

はじめに.....	3
NVIDIA Tegra 4 ファミリのモバイル・プロセッサ	4
CPU 性能のベンチマーク	4
高性能、高電力効率を重視して設計された Tegra 4 ファミリの CPU	7
より広い発行実行ユニットによるスループット向上.....	7
アウトオブオーダー実行の命令ウィンドウを広げてメモリ・レベルの並列性を向上	8
高速の Load-To-Use ロジックによる L1 データ・キャッシュの容量拡大.....	9
分岐予測機能の強化による効率向上	10
高度なプリフェッチャによる MLP 向上とレイテンシ低減	11
大容量の統合型 L2 キャッシュ	12
電力効率を高めた Cortex-A15	13
第 5 のバッテリーセーブ型 CPU コア.....	15
まとめ.....	Error! Bookmark not defined.
付録.....	18
文書改定履歴.....	20

はじめに

NVIDIA は 2011 年初めに業界に先駆けて初のマルチコア・モバイル SoC (システム・オン・チップ) プロセッサ「**Tegra® 2**」を導入し、モバイル・コンピューティングを一変させました。世界初のデュアルコア・モバイル・プロセッサである Tegra 2 は、タブレットやスマートフォンに搭載されて出荷開始され、パフォーマンス、バッテリー動作時間、ユーザ体験の向上を可能にしました。ハイエンド・スマートフォンやタブレットの急速な普及とともに、これらのデバイスはますます、消費者が日々の情報処理や娯楽ニーズを満たすために不可欠なものとなっています。

そして、Tegra 3 の導入により、NVIDIA は世界初の **4-PLUS-1™** クアッドコア・モバイル・プロセッサの出荷を開始しました。この類例のない 4-PLUS-1 アーキテクチャは、4 個の CPU コアを高い処理性能を要求するアプリケーションに使用し、1 個の**バッテリーセーブ型コア**を低い処理性能で済むアプリケーションや、デバイスのアイドル時のバックグラウンド・タスク処理に使用します。他に類を見ないこの組み合わせは、消費者にパフォーマンス向上を提供し、本格的な Web ブラウジングやコンソールクラスのゲーミング、高度なマルチタスキング、写真・動画編集を可能し、それと同時にバッテリー動作時間も向上させます。

次世代のマルチコア・モバイル・プロセッサは大幅な性能向上が見込まれており、スマートフォンやタブレット上で、PC 並みの生産性やソーシャルネットワーク、マルチメディア、コンソール品質のゲーミング体験が可能になると予想されています。また、オペレーティング・システムはモバイル・プロセッサの高まる性能を、新機能やより優れた機能の提供、ユーザ・インターフェースの高速化・効率化・表現力向上、アプリケーション・パフォーマンスの向上に利用し続けています。

Tegra 4 のようなパワフルな SoC デバイスの導入により、スマートフォンやタブレット、ノート PC の処理能力は接近しつつあり、モバイル端末で PC クラスのパフォーマンスと妥協のないバッテリー動作時間が実現することに消費者の期待が高まっています。より高解像度のディスプレイと Miracast™ などのワイヤレス・ディスプレイ・ソリューション、そしてコンソールクラスのモバイル・ゲームが普及することで、消費者はタブレットやスマートフォンを、ポータブル・ゲーム端末と家庭用ゲーム機の両方として使用するようになるでしょう。

NVIDIA の最新 SoC プロセッサ「**Tegra 4** ファミリ」のアーキテクチャは、この新しい一群のモバイル用途の非常に高い要求仕様に対応すべく設計されています。

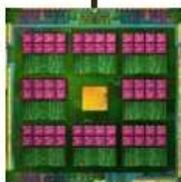
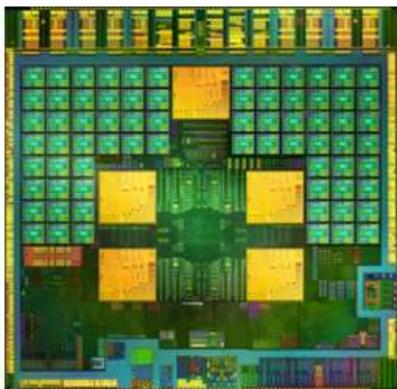
NVIDIA Tegra 4 ファミリのモバイル・プロセッサ

NVIDIA Tegra 4 は、**ARM Cortex-A15** を採用した世界初のクアッドコア CPU アーキテクチャであり、要求の厳しいモバイル・アプリケーションやバッテリー動作時間の向上に対応できる優れたパフォーマンスを提供します。また、Tegra 4 は、第 2 世代の**バッテリーセーブ型 CPU コア**と**バリエابل対称型マルチプロセッシング (vSMP)** 技術の採用によって、パフォーマンスとバッテリー動作時間が一段と向上しました。

Tegra 4 は Tegra 3 に比べて動作速度が大幅に速く、電力効率も高いことから、より高速なパフォーマンスと、より表現豊かなグラフィックスを、ほぼ同じ電力消費で提供できます。

Tegra 4

世界最速のモバイル・プロセッサ



Tegra 4i

初の統合型Tegra 4 LTEプロセッサ

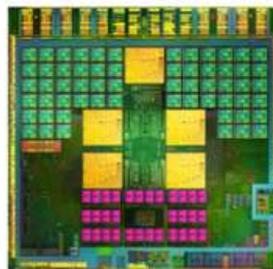


図 1 NVIDIA Tegra 4 ファミリ

NVIDIA Tegra 4i は Tegra 4 アーキテクチャを採用しています。Tegra 4i は、Tegra 4 を搭載したスーパー・フォンのモバイル体験を、普及価格帯のスマートフォンに取り込みます。これを可能にする各種機能と NVIDIA の LTE i500 SDR モデムが、すべてワンチップに集積されました。Tegra 4i は ARM の最新かつ最も高効率の CPU コア「**Cortex-A9 r4**」をクアッドコア実装(動作クロック 2.3GHz)し、第 5 のバッテリーセーブ型コアを搭載しています。サイズは競合品のクアッドコア統合型 LTE モバイル・プロセッサの半分に抑えられています。

Tegra 4i に搭載された最新 CPU は、NVIDIA の支援を得て ARM が設計したものです。この CPU は、効率とパフォーマンスの新たな基準を打ち立てるとともに、Tegra 3 に搭載されている前リビジョンの Cortex-A9 に比べて 2 倍の性能を達成しています。Tegra 4i は超高速の Web ブラウジングと速いアプリのロード時間を、余裕のあるバッテリー動作時間とともに実現しています。

CPU 性能のベンチマーク

ベンチマークは一般的に、実世界における性能とベンチマーク結果との間に良好な相関が得られる形で、デバイスや設計の性能を評価するために使用されます。そのため、CPU アーキテクチャの性能を

評価する場合には、CPU 性能のベンチマーク評価に使用する負荷を、実世界のアプリケーションやプログラムによる負荷と同程度のものにすることが不可欠です。

これまでに、CPU の性能を評価するために各種のベンチマークが開発されてきました。CPU のベンチマーク評価に使用されてきた有名なベンチマークに、Dhrystone MIPS と Coremark の 2 つがあります。Dhrystone MIPS (DMIPS と呼ばれる) は、約 30 年前に、主として CPU の整数演算性能を示す合成ベンチマークとして開発されました。

残念ながら、DMIPS は、CPU アーキテクチャが現在よりも遙かに簡素で、外部メモリの速度と緊密に関係していた 30 年前には有効でしたが、今では時代遅れとなり、実世界のアプリケーションに要求されるパフォーマンスを正確に反映しなくなりました。DMIPS には評価指標として大きな問題があります。それはベンチマーク・コードが現在のモバイル・プロセッサの L1 キャッシュにすっぽりと収まってしまうことです。そのために、L1 キャッシュのミス・ハンドリング・ロジックや L2 キャッシュ、SoC のメモリ・システムの処理性能を評価できません。Dhrystone ベンチマークの開発者ははるか以前の 1999 年に次のように述べています。

「DMIPS はあまりにも短くてオンチップ・キャッシュに収まってしまい、メモリ・システムに十分な負荷をかけることができないため、近年のプロセッサ処理負荷を扱うのに有用とは言えない」

DMIPS の開発者 (Weicker)、EDN Magazine、1999 年 10 月 28 日

同様に、Coremark や Linpack など、その他の広く用いられているベンチマークは、CPU 性能のある特定の部分のみをテストするために書かれています。Coremark は ALU と FPU の性能のみをテストするのに対し、Linpack は CPU の浮動小数点ユニットの性能の評価尺度にすぎません。

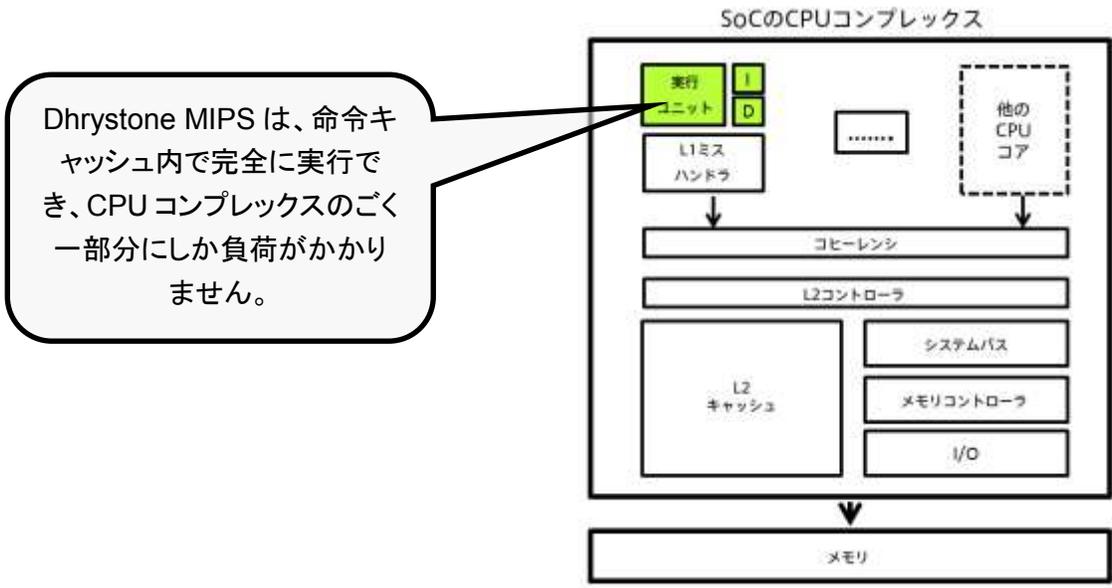


図2 Dhrystone MIPS は CPU コンプレックスのごく一部だけをベンチマーク評価

真の CPU 性能をより正確に表すベンチマークに、SPECint ベンチマークがあります。SPECint は、合成テストを使用するのではなく、ファイル圧縮やワープロ、ゲーミング、データベース管理など実世界の各種アプリケーションのカーネルを使用することを特に想定して書かれたものです。

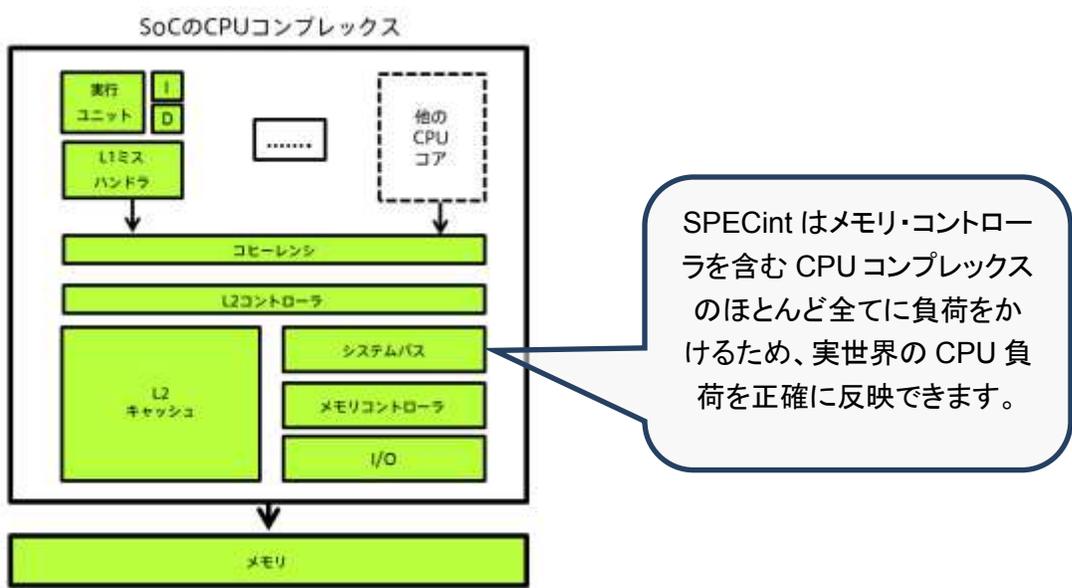


図3 SPECint は CPU コンプレックス全体に負荷をかけ、負荷は実世界のアプリケーションに類似

高性能、高電力効率を重視して設計された Tegra 4 ファミリの CPU

現世代のモバイル・アプリケーションは、現在のモバイル・プロセッサのパフォーマンス能力を押し上げています。マルチコア CPU アーキテクチャと、より高い動作周波数への移行によって、プロセッサの処理能力は向上し、高まる性能ニーズに対応できるようになりました。しかし、次世代の PC クラスのモバイル・アプリケーションに必要な CPU アーキテクチャは、高まる性能ニーズに対応するのに十分な性能上の余裕があるだけでなく、消費電力がモバイルの電力バジェット(許容消費電力)内に常時収まる新しい CPU アーキテクチャです。

Tegra 4 の Cortex-A15 CPU アーキテクチャと、Tegra 4i の Cortex-A9 r4 CPU アーキテクチャはともに、現世代のアーキテクチャに比べて改良点がいくつかあり、それぞれのデバイス・クラスの現行ソリューションと比較して最大 2 倍の性能向上を実現しています。以下、主な改良点をいくつか説明します。

より広い発行命令数の実行ユニットによるスループット向上

Cortex-A15 CPU アーキテクチャは 8 つの実行ユニットを有し、1 クロックあたり 8 つの命令を発行できます(Cortex-A9 は 5 つ)。しかし、有効に活用できない限り、実行ユニットはほとんど無価値です。数が増えた実行ユニットからメリットをフルに引き出すためには、アーキテクチャは、より深さのあるアウトオブオーダー実行に対応して命令レベルの並列性(ILP: Instruction Level Parallelism)を高めるとともに、キャッシュミスの早期検出によりメモリ・レベルの並列性(MLP: Memory Level Parallelism)を高める必要があります。また、動作効率を高めるために正確な分岐予測機能も備えている必要があります。

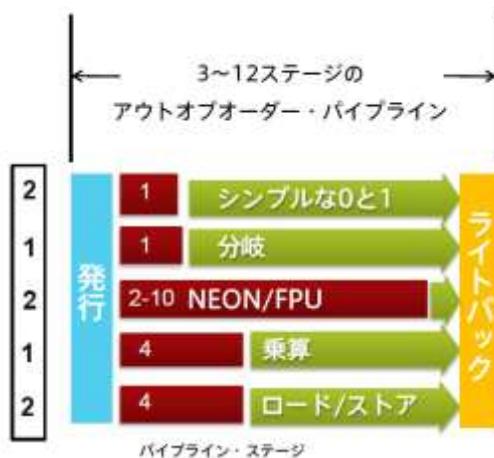


図 4 ARM Cortex-A15 CPU に搭載されている 8 つの実行ユニット

アウトオブオーダー実行の命令ウィンドウを広げてメモリ・レベルの並列性を向上

CPU パフォーマンスは、メモリアクセスのレイテンシ削減に大きく依存しています。メモリからのデータ・フェッチを待って CPU がアイドル状態になっていたのでは、CPU のパフォーマンス効率は大幅に低下してしまいます。ですから、より大容量かつ効果的な CPU キャッシュを使ってキャッシュミス(これがメモリからのデータ・フェッチをトリガ)の数を減らすだけでなく、メモリ・レベルの並列性(MLP:システム・メモリからのデータ・フェッチの同時実行数)を高めることも重要です。システム・メモリから CPU へのデータ・フェッチの実効レイテンシは、メモリ・フェッチ 1 回のレイテンシを、並列実行できるメモリ・フェッチの数(MLP)で割り算した値に等しくなります。

ですから、CPU のパフォーマンス効率を高めるためには、ずっと先の命令まで見通して、キャッシュでミスする可能性のあるロード命令を発行することが重要です。アウトオブオーダー・ルックアップ・ウィンドウが広いほど、そのようなロードを見つけられる可能性が高まります。

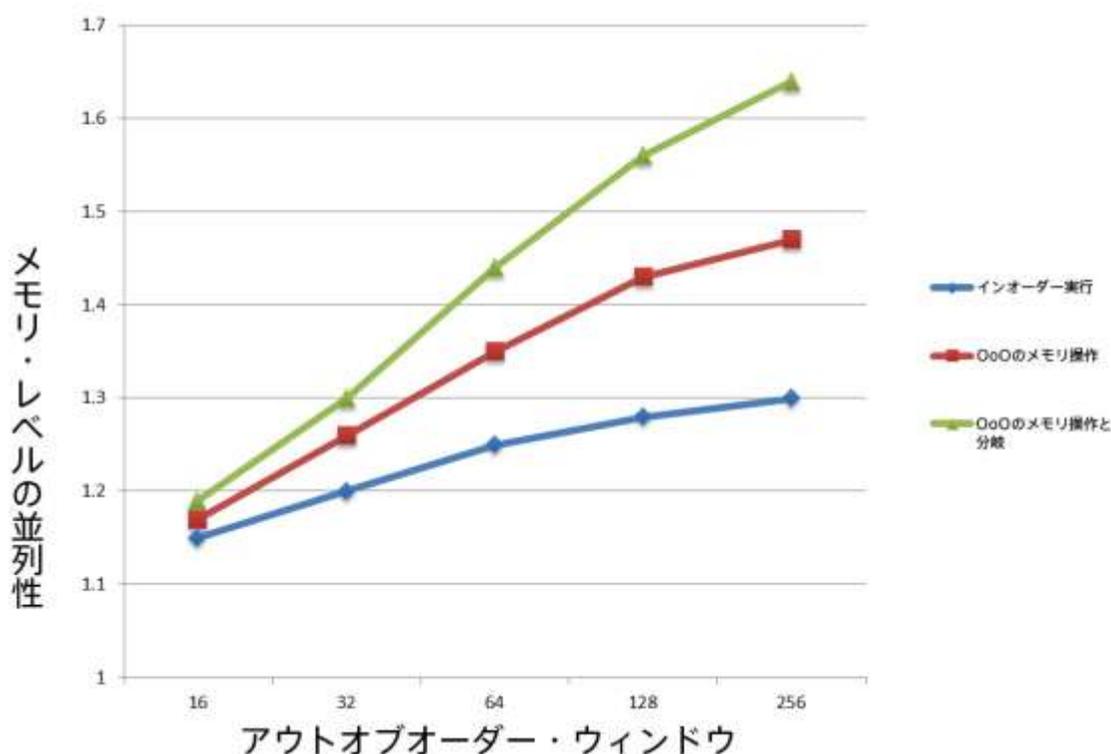


図 5 メモリ・レベルの並列性と OoO ウィンドウのルックアップ・サイズの関係

図 5 は、より広い OoO ウィンドウを使用する際に CPU アーキテクチャ強化の重要性を示しています。アプリケーションにおいて得られる MLP を最大化するためには、メモリ操作と分岐両方のアウトオブオーダー実行が必要です。

Cortex-A15 アーキテクチャは、128 個のマイクロ命令 (micro-op) を扱える広いルックアップ・ウィンドウを採用して MLP を大幅に高め、分岐と一部のメモリ操作をアウトオブオーダー実行します (ARM 命令の大半には、命令とマイクロ命令の間に一対一のマッピングがありますが、より複雑な CISC 的な命令の一部には 2 つまたはそれ以上のマイクロ命令に展開可能です)。これに対し、クアルコム の Krait プロセッサ・アーキテクチャは 40 命令の OoO ウィンドウしかなく、MLP の向上余地が限られています。

なお、すべてのメモリ操作はアウトオブオーダーで完了できますが、OoO によるメモリ操作発行における唯一の制約は、より古いストアについてはアドレスが分かっている点です。そのため、A15 は上掲のグラフでは、フルに OoO 実行した場合に当たる緑色の線の近くとなります。

高速の Load-To-Use ロジックによる L1 データ・キャッシュの容量拡大

現在の CPU 設計におけるクリティカル・パスの一つは、ある操作が、先行するロードの結果に依存する、load-to-use パスです。このパスには、次のようないくつかの複雑なステップがあります。

- アドレスを計算するために加算
- 仮想アドレスを物理アドレスに変換
- アドレスをタグの内容と比較
- 得られたキャッシュ・ラインからクリティカル・データを選択
- そのデータを必要なレジスタにアライン

Cortex-A15 は、極めて効率的な論理設計を用いて加算ステップと変換ステップの論理を組み合わせています。この論理設計は、完全な (しかし時間のかかる) 加算演算を行わずに、比較がヒットするかどうかの判定に十分なだけのデータを生成します。これらのステップを組み合わせることで、load-to-use パスの速度が上がり、高速の 32KB L1 データ・キャッシュの実装が可能になります。一般には、大容量の L1 キャッシュは、小容量のキャッシュに比べて低いクロック速度で動作する必要があります。実際、Krait などの競合アーキテクチャは、16KB の L1 データ・キャッシュのサイズに制限されています。図 6 はデータ・キャッシュのサイズが SPECint の評価結果に与える影響を示しており、32K データ・キャッシュの重要性を示しています。

Cortex-A9 r4 は同じ論理と 32K キャッシュを組み合わせしており、また、TLB のサイズを A15 と同じ 512 エントリに増量しています。

- Cortex-A15 と Cortex-A9 r4 は、極めて効率的な load-to-use パスに対応しています。
- Cortex-A15 と Cortex-A9 r4 は、32KB の L1 データ・キャッシュを備えています。
- Cortex-A9 r4 も、Cortex-A15 と同じ大容量の TLB (変換ルックアップ・バッファ) を備えています。

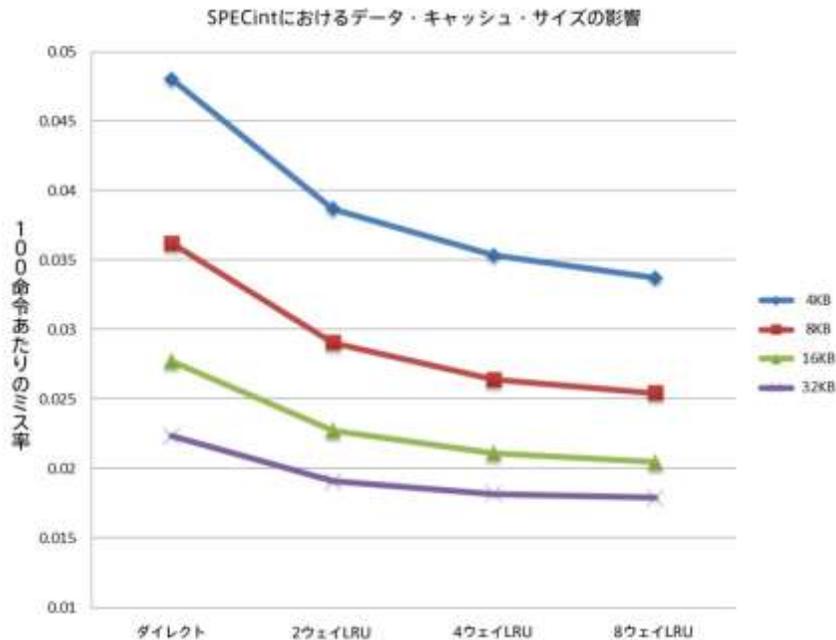


図 6 データ・キャッシュのサイズがキャッシュミス率と CPU パフォーマンスに与える影響

分岐予測機能の強化による効率向上

正確な分岐予測は、OoO 実行の効率を確保する上で、MLP を引き出すためにも、無駄な動作を回避するためにも極めて重要です。Cortex-A15 は、次の機能を備えた高度な分岐予測メカニズムを備えています。

- 反対方向に向かうエイリアシング分岐間の干渉を軽減するための 2 モード予測器。2 モード予測器の詳細については、インターネット上に一般的な説明があります。
- 分岐を OoO で解決して、推測ミスをより早期に訂正し、無駄な動作を回避します。
- 投機的なポップ (pop) やプッシュ (push) からのリターン・アドレス・スタックの訂正機能。A15 のような積極的なアウトオブオーダー・マシンは、投機的なコールやリターンでリターン・アドレス・スタックを損なう恐れがあり、そのため、推測ミスを訂正するメカニズムを備えています。
- 間接分岐予測メカニズム。間接分岐は Web 言語に要求されるインタプリタによく使われています。間接分岐をより正確に予測するために、A15 には、PC と履歴をもとにターゲットを見つけるためにアクセスされる構造が別途備わっています。
- BTB のコールド・ミスを予測するために静的予測メカニズムが必要です。

Cortex-A9 r4 アーキテクチャは、A15 と同じ大容量の履歴レジスタを有し、予測器構造のサイズも同じです。これらの改良により、Cortex-A9 r4 CPU は Cortex-A15 アーキテクチャと同程度の予測精度を提供できます。

- Cortex-A9 r4 は、履歴バッファの拡大と飽和カウンタにより、Cortex-A15 並みの分岐予測性能を達成しています。

高度なプリフェッチャによる MLP 向上とレイテンシ低減

MLP を高めるために先進プロセッサに採用されるもう一つの方法は、ハードウェア・プリフェッチャです。プリフェッチャは、必要になるデータを事前に予測することで、メモリ・レイテンシを下げ、MLP を高めることができます。特に、アウトオブオーダー・マシンがなかなか検出できない状況があります。たとえば、プログラムがメモリを介して規則的な構造を持つ連結リストをたどっている場合、先行するロードが完了してアドレスが生成されるまで、それに続くロードの発行ができません。ストライドを検知できるプリフェッチャは、こうしたデータを事前にロードすることができます。

Cortex-A15 は L2 コントローラに、1 CPU あたり 1 個のハードウェア・プリフェッチャを搭載しています。重要な点として、このプリフェッチャはプログラム・カウンタ(PC)とアドレス・ビットの併用して、ストライド・アクセス(等間隔アクセス)を検知し、PC 不使用の場合なら発生していた異なるスレッド間のエイリアシングを回避します。

実行するプリフェッチの回数はプログラム可能であり、リソース浪費を回避するためのハードウェア・スロットル・メカニズムも備わっています。

Cortex-A9 r4 は各 CPU に L1 プリフェッチャが新設されており、同じく PC とアドレスを併用します。このプリフェッチャは専用バッファを持ち、データがヒットするまで L1 データ・キャッシュへの割り当てを行いません。また、有効性を自己チェックして、有効でない場合にはバックオフするほか、複数のストリームを追跡できます。

- Cortex-A9 r4 は、L1 データ・プリフェッチャを備えています。
- Cortex-A9 r4 のプリフェッチャは、専用のプリフェッチ・バッファを持ち、複数のプリフェッチ・ストリームを追跡できます。

大容量の統合型 L2 キャッシュ

キャッシュは、システム・メモリへのオフチップ・アクセスの回数を減らすために、CPU で使用されます。キャッシュは使用頻度の特に高いデータをチップ上に格納することで、CPU がより高速にデータにアクセスできるようにし、CPU のパフォーマンスや効率を高めます。NVIDIA Tegra 4 に搭載されているクアッドコア ARM Cortex-A15 CPU コンプレックスの各コアは、それぞれ 32KB の命令キャッシュと 32KB のデータ・キャッシュを搭載しています。4 個のコアが共通の 2MB の大容量 L2 キャッシュを共有します。これは 16 ウェイ・セット・アソシアティブです。128 エントリの深さをもつ大容量のアウトオブオーダー・バッファにより、L2 キャッシュ・レイテンシは概ね隠蔽できます。32KB の L1 キャッシュとともに、2MB の L2 キャッシュはシステム・メモリからのオフチップ・フェッチを最小限に抑える機能を果たします。DRAM からのフェッチは内蔵 SRAM からのフェッチに比べて消費電力が多いことから、これはパフォーマンス向上と消費電力の低減の両方につながります。

クアッドコア Cortex-A15 アーキテクチャは、クアルコムの Krait アーキテクチャとは L2 キャッシュの実装がまったく異なります。Krait とは異なり、NVIDIA Tegra 4 では搭載されている 2MB キャッシュ全体が、4 個のコアすべてに完全共有されます。各コアに割り当てられる容量は、負荷に応じてハードウェア制御により動的に完全に変化させることができます。たとえば、Tegra 4 では、1 個のコアに 1280KB を割り当てる一方で、他の 3 個のコアにそれぞれ 256KB ずつを割り当てて合計を 2MB とすることも可能です。同様に、マルチスレッド・アプリケーションが共通のデータ構造を共有していれば、4 個のコアすべてが同一の L2 キャッシュ・ラインにアクセスすることもできます。このような負荷ベースによる L2 キャッシュ容量の割り当ては、L2 キャッシュの効率を高め、キャッシュミス率のさらなる低減に寄与します。

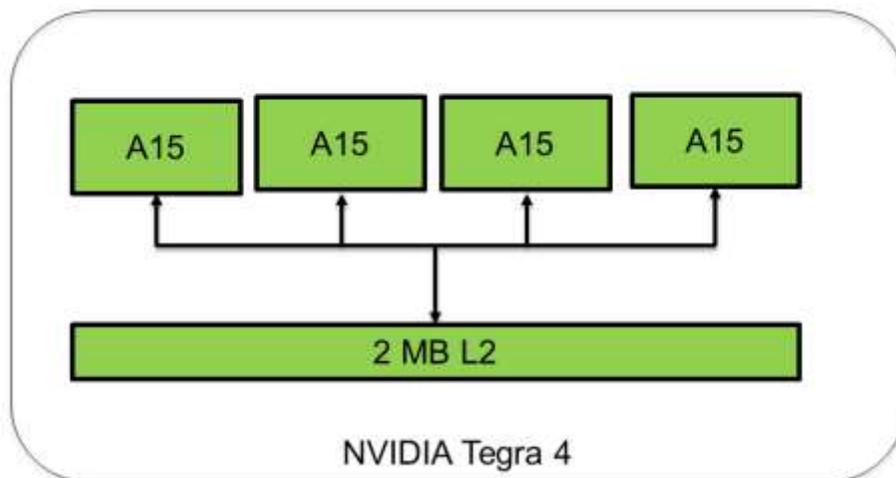


図 7 NVIDIA Tegra 4 の任意の CPU に、2MB の L2 キャッシュ全体を動的に割り当て可能

これに対し、Krait は、非同期 SMP アーキテクチャの限界があるため、各 CPU コアに 512KB という固定された量の L2 キャッシュを静的に割り当てます。Krait の CPU コアの 1 つが 512KB を超えるデータを使用すると、必然的に L2 キャッシュミスが増え、システム・メモリからのデータ・フェッチが行われて高くつきます。より容量の小さい 512KB のキャッシュの静的割り当てが、Krait のパフォーマンスに与

える影響は、実世界のアプリケーション・パフォーマンスを模擬した、SPECint2Kなどの CPU ベンチマークで容易に確かめられます。

さらに、スレッドがコアからコアへ移動する際、移動先のコアは他のコアの L2 からデータをフェッチし、ローカルな L2 に移動させる必要があります。非同期 SMP の境界上では各コア間で異種クロック・ドメイン間の同期化が必要となるため、このコア間トラフィックは大きな追加遅延を被る可能性があります。

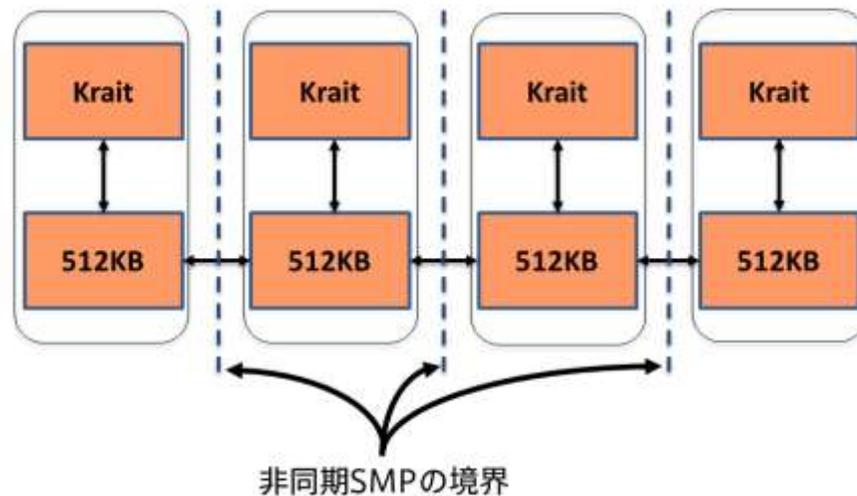


図 8 Krait は非同期 SMP のため、L2 全体を 1 個の CPU コアに割り当てることは不可

電力効率を高めた Cortex-A15

よりハイレベルなパフォーマンスを提供する各種機能に加えて、Cortex-A15 CPU アーキテクチャには、従来のアーキテクチャに比べて電力効率の向上を可能にするアーキテクチャの改良点がいくつかあります。

A15 アーキテクチャは、アクティブな電力管理を行うために大容量の 32 エントリのループ・バッファを実装しています。Cortex-A15 は、命令フェッチ、命令デコード、分岐予測の 3 ブロックで CPU の消費電力の約 40%を消費します。CPU がループ・バッファから実行している時には、この 3 ブロックは停止され、クロック・ゲーティングでオフにされます。下図をご参照ください。

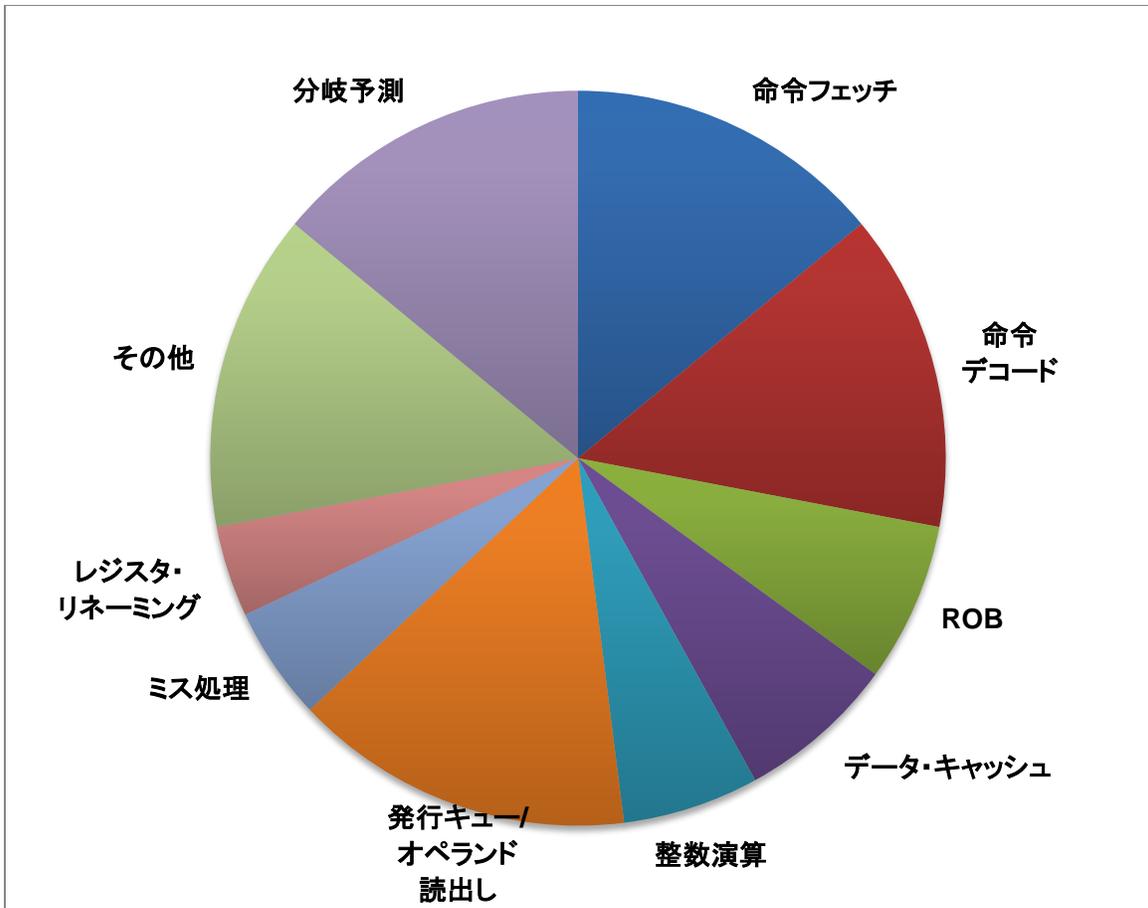


図 9 深いループ・バッファにより、CPU コアの消費電力(フロントエンド)のほぼ 50%を節約

Cortex-A15 は、消費電力が増えすぎてパフォーマンス向上がモバイル負荷に見合わなくなるケースでは、アウトオブオーダー命令発行を制限することで、パフォーマンスと消費電力のバランスをとります。たとえば、ロード命令はより古いストア命令の前には発行されず、新しいストア命令はより古いストア命令が実行される前には発行されません。これにより消費電力の多いメモリ・ディスアンビギュエーション(非曖昧化)構造を回避できます。

また、L1 および L2 キャッシュを含む設計全体にわたって本格的なクロック・ゲーティングとパワー・ゲーティングが実装されており、いっそう消費電力を低減します。

第 5 のバッテリーセーブ型 CPU コア

4 個の高性能 CPU コアに加えて、NVIDIA Tegra 4 ファミリのクアッドコア CPU アーキテクチャは、第 5 の低電力バッテリーセーブ型 CPU コアを搭載しています。このアーキテクチャに採用された第 5 のバッテリーセーブ型 CPU コアは、低消費電力重視で最適化されており、動作に必要な電力の少ないトランジスタを使って構築されています。このバッテリーセーブ型コアは電力効率が抜群に高く、バックグラウンドでのメール同期やソーシャルメディアとの同期、オーディオ再生、ビデオ再生、電子ブックなど、パフォーマンス要求の低い各種タスクの処理に焦点が当てられています。

このバッテリーセーブ型コアは低パフォーマンス・タスクを扱うことを想定して設計されています。これらのタスクは一般的に、2MB の L2 キャッシュをフルに利用しません。さらなる電力節減のため、バッテリーセーブ型コアにはそれ自体の 512KB L2 キャッシュが備わっています。メイン CPU コアがオフになると、2MB L2 キャッシュはフラッシュされ、電力節減のためにパワー・ゲーティング機能が働きます。バッテリーセーブ型コアはより小容量で消費電力の少ない 512KB L2 キャッシュを使って動作します。

高性能クアッドコア A15 CPU コンプレックスの使用が、パフォーマンス要件に照らして正当化されないケースでは、Tegra 4 はメイン・クアッド CPU コンプレックスを完全にオフにし、バッテリーセーブ型コアに切り替えます。メイン・クアッドコア A15 CPU コンプレックスとバッテリーセーブ型 CPU コアの切り替えは、NVIDIA の第 2 世代のバリエーション対称型マルチプロセッシング (VSMP) 技術によって管理されます。この技術は、非同期 SMP 技術を採用しているクアルコム の Krait アーキテクチャに比べて高水準の電力効率を提供します。

vSMP 技術を採用した NVIDIA Tegra 4 の 4+1 クアッドコア・プロセッサには、下記のようなアーキテクチャ上の大きな利点があります。

- **キャッシュ・コヒーレンス**: vSMP 技術はバッテリーセーブ型コアとメイン・パフォーマンス・コアが両方同時に有効化されることを認めませんから、異なる速度で動作しているコア間でのキャッシュ同期化に伴うペナルティはありません。これに対し、ASMP の場合、より高い周波数で動作しているコアが、それよりもずっと低い周波数で動作しているコアとのデータ同期化を試みる際に、キャッシュ同期化に伴うペナルティを被ります。これはより高い周波数で動作しているコアのパフォーマンスに影響を与えるだけでなく、コア間に同期化ハードウェアが追加されることにもなり、消費電力が増加します。

さらに、非同期クロッキングに対応するために、Krait アーキテクチャは L2 キャッシュの静的割り当てを採用しており、各 CPU コアの利用可能な L2 キャッシュ容量はわずか 512KB に制限されています。利用可能なキャッシュ容量の少なさは、L2 キャッシュのヒット率に影響を与えるとともに、オフチップのメモリ・フェッチ増加とメモリ・レイテンシの増大による CPU パフォーマンスへの影響によって消費電力の増加を招きます。

OS の効率: Android OS と WinRT OS は、すべての利用可能な CPU コアが同一で、同程度の処理性能を備えていると仮定しています。この仮定に基づいて、OS は複数の CPU コアにタスクをスケジューリングします。複数の CPU コアがそれぞれ異なる非同期の周波数で動作し

ている場合、コアのパフォーマンス能力に差ができることになります。これは OS のスケジューリングの非効率化につながる恐れがあります。これに対し、vSMP 技術では、常にすべてのアクティブ・コアを同程度の同期した動作周波数に保つことで、OS のスケジューリングの最適化を図ります。vSMP がバッテリーセーブ型コアから 1 個または複数のメイン CPU コアに切り替わる場合にも、CPU の管理ロジックによって、エンドユーザには知覚されないものの、OS のスケジューリング上のペナルティを招かないシームレスな移行が確実に行われます。

NVIDIA Tegra の 4+1 CPU アーキテクチャと vSMP 技術についての詳細は、ホワイトペーパー「[Variable SMP – A Multi-Core CPU architecture for High Performance and Low Power](#)」(vSMP 技術—高性能・低消費電力を重視したマルチコア CPU アーキテクチャ) (英語)をご参照ください。

まとめ

スマートフォンやタブレットは、パーソナル・コンピューティング・デバイスとして普及が進んでいます。現在のスマートフォンはもはや、携帯電話やメッセージング、軽い Web ブラウジングなどに用途が限られません。写真編集やワープロ、マルチタブの Web ブラウジング、最新のグラフィック・リッチなゲーミング、マルチタスキングといった、PC クラス用途向けのモバイル・アプリケーションが、モバイル端末向けに登場しており、モバイル端末のパフォーマンス要件を押し上げています。NVIDIA Tegra 4 ファミリのモバイル SoC に搭載されているクアッドコア CPU は、次世代のモバイル・アプリケーション向けにより高いパフォーマンスを提供できるよう、いくつかの重要な機能強化がなされています。Tegra 4 のクアッドコア Cortex-A15-CPU コンプレックスは、命令レベルの並列性を高め、アウトオブオーダー・ウィンドウを広げ、分岐予測をより高機能化するために実行リソースを充実させ、レイテンシの低減と効率向上を図りました。また、データ・アクセスや命令アクセスの全般的な速度を高め、キャッシュミスに伴うペナルティを削減できるよう、大容量のデータ・キャッシュと、より大容量の L2 キャッシュを備えています。

Tegra 4i に搭載されているクアッドコア Cortex-A9 r4 プロセッサは、履歴バッファの拡大や TLB サイズの増量、プリフェッチャ・ユニットの搭載による分岐予測の改良をはじめ、Cortex-A15 クラスの改良がいくつか取り入れられており、標準的な Cortex-A9 アーキテクチャを大幅に上回るパフォーマンスと電力効率を提供します。

これらの改良の利点は、ワープロやゲーミング、データベース、ファイル圧縮など実際のアプリケーションで発生する負荷を模擬して設計された、業界に普及している SPECint2000 のベンチマーク結果を見れば容易にわかります。SPECint2000 やその他のベンチマークの結果は、Tegra 4 ファミリのクアッドコア CPU アーキテクチャが、次世代のモバイル・アプリケーションで予想される負荷に対して、競合アーキテクチャに比べて高パフォーマンスを発揮できることを明確に示しています。

しかも、Tegra 4 ファミリのプロセッサは電力効率が抜群に優れており、前世代の Tegra 3 プロセッサと比較しても同一負荷条件時の消費電力は 30% 少なく抑えられています。¹

¹ SPECint2000 でのパフォーマンス・レベルを一定として、Tegra 3 と Tegra 4 の各開発プラットフォーム上での SoC+メモリの消費電力を比較。

付録

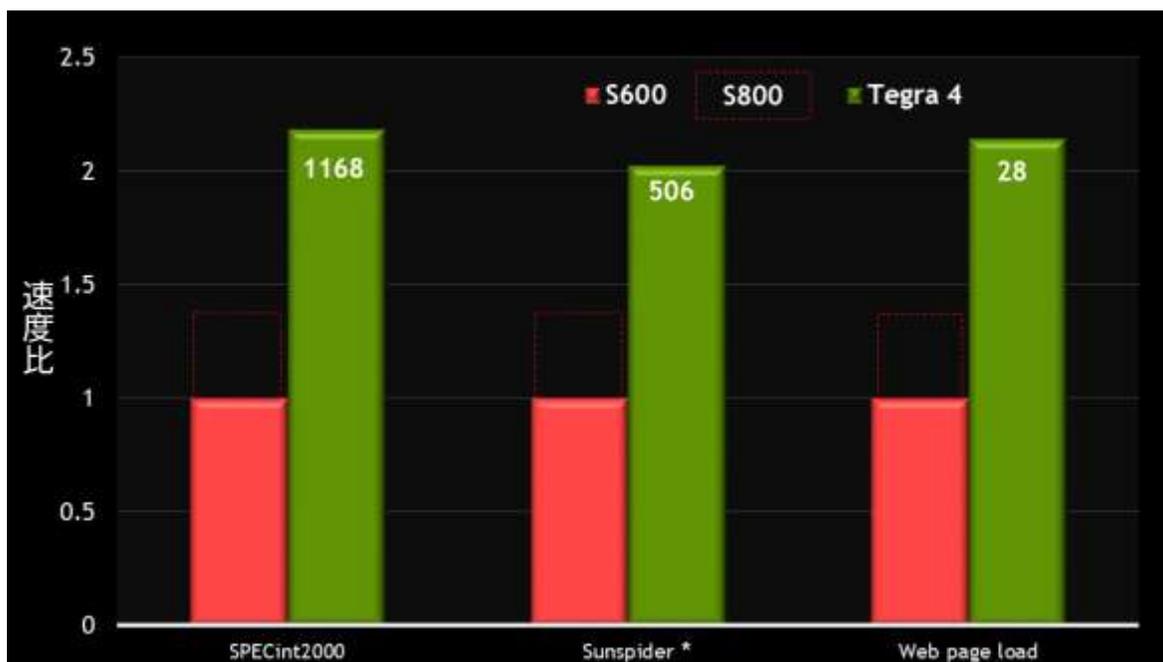


図 10 CPU および Web ベンチマークによる Tegra 4 と競合製品の性能比較²

	ベンチマーク	スコア
CPU/システム	SpecINT2000	1168
	Sunspider 0.91	506 ミリ秒
	Web Page Load	28 秒
	WebGL Aquarium(50 匹)	60fps
	Google Octane	4582
	Kraken 1.1	6799 ミリ秒
	Geekbench 1.0	4285
	Antutu 3.1.1	36127
	Quadrant Pro 2.0	16449
	CFBench 1.1	41227
GPU	GLBench 2.5 HD Egypt(オフスクリーン 1080p)	57fps
	GLBench 2.5 HD Classic(オフスクリーン 720p)	274fps
	Basemark ES 2 Hoverjet	59fps

表 1 Tegra 4 のベンチマーク評価結果

² Tegra 4 のスコアは Tegra 4 開発プラットフォーム上で計測。Sunspider の S600 のスコアは公表値を使用。その他のスコアは市販の S4 Pro を使った計測値から S600 の数値を推定。S800 のスコアは、クアルコムが発表している、より高いクロック周波数と IPC をもとに S600 の数値から推定。

文書改定履歴

改訂番号	摘要

注記

解説、見解、NVIDIA のデザイン仕様、リファレンスボード、ファイル、図面、診断、リスト、その他のドキュメントなど、本ホワイトペーパーで提供する情報は(以下、集成的あるいは個別に「マテリアル」と呼称します)、すべて、「現状のまま」を条件に提供するものです。NVIDIA は、本マテリアルについて、明示的、暗示的、法定的を含む保証を一切行わず、また、権利非侵害、商品性、および特定目的適合性に関するあらゆる黙示保証を明示的に否認します。

記載された情報の正確性、信頼性には万全を期しておりますが、これらの情報の使用の結果に対して、もしくはこれらの情報の使用に起因して第三者の特許またはその他の権利の侵害が発生しても、NVIDIA Corporationは一切の責任を負わないものとします。暗示的か明示的かを問わず、NVIDIA Corporationの特許または特許権に基づいて付与されるライセンスは一切ありません。本文書に記載された仕様は予告なしに変更されることがあります。本文書は、過去に提供されたすべての情報に優先し、これに代わるものとします。NVIDIA Corporationの製品は、NVIDIA Corporationの書面による明示的な許可なくしては、生命維持装置の重要な部品として使用してはなりません。

商標について

NVIDIA、NVIDIAのロゴ、Tegraおよび4-PLUS-1は、米国およびその他の国におけるNVIDIA Corporationの商標または登録商標です。その他の会社名および製品名は各社の商標である可能性があります。

Copyright

© 2013 NVIDIA Corporation. All rights reserved.