

この資料は英語版を翻訳したもので、内容に相違が生じる場合には原文を優先します。こちらの日本語版は参考用としてご利用ください。設計の際には、最新の英語版で内容をご確認ください。

NII51018-5.1.0

### はじめに

Nios® II 開発キットは、リリースごとに Nios II プロセッサまたはキットの開発ツール、あるいはその両方が改良されています。本書では、Nios II プロセッサのリビジョン履歴を記載しており、Nios II IDE などの開発ツールのリビジョンには触れていません。

Nios II プロセッサの改良により、以下が影響を受けることがあります。

- Nios II アーキテクチャの機能 — アーキテクチャ・リビジョンの一例は、浮動小数点演算をサポートする命令の追加です。
- 特定の Nios II コアの実装 — コア・リビジョンの一例は、Nios II/f コアに対するデータ・キャッシュ・メモリの最大許容サイズの増大です。
- JTAG デバッグ・モジュールの機能 — JTAG デバッグ・モジュール・リビジョンの一例は、JTAG デバッグ・モジュールへの追加トリガ入力の追加です。これにより、新しいタイプのトリガ・イベントでプロセッサの実行を停止できます。

アルテラは、既存の Nios II コア用に記述されたコードが同じコアの将来のリビジョンでも動作するように、Nios II リビジョンを実装します。

### Nios II の バージョン

Nios II プロセッサのバージョン番号は、Nios II 開発キットのバージョンによって決定されます。例えば、Nios II 開発キットのバージョン 1.01 では、Nios II コアのバージョンもすべて 1.01 になります。

表 6-1 に、Nios II 開発キットの全リリースのバージョン番号を示します。

バージョン	リリース月	説明
5.1	2005 年 10 月	変更なし。
5.0	2005 年 5 月	<ul style="list-style-type: none"> <li>● バージョン名を変更。Nios II プロセッサのバージョンとアルテラの Quartus II® ソフトウェアのバージョンが一致。</li> <li>● メモリ構造の強化：               <ol style="list-style-type: none"> <li>(1) 密結合メモリ (Tightly Coupled Memory) を追加。</li> <li>(2) データ・キャッシュ・ライン・サイズをコンフィギュレーション可能に更新。</li> <li>(3) Nios II/f および Nios II/s コアのキャッシュをオプションに更新。</li> </ol> </li> <li>● ハードウェアによって Cyclone™ II デバイス・サポートを確認。</li> <li>● HardCopy® デバイスをサポート。</li> </ul>
1.1	2004 年 12 月	<ul style="list-style-type: none"> <li>● アーキテクチャのマイナーな機能拡張: cpuid コントロール・レジスタを追加し、break 命令を更新。</li> <li>● Nios II/s および Nios II/f コアの論理演算ユニット (ALU) で、乗算およびシフト・ハードウェアに対するユーザ制御の強化。</li> <li>● マイナーなバグの修正。</li> </ul>
1.01	2004 年 9 月	<ul style="list-style-type: none"> <li>● ハードウェアでの Stratix™ II デバイス・サポートを確認。</li> <li>● マイナーなバグの修正。</li> </ul>
1.0	2004 年 5 月	Nios II プロセッサの初回リリース。

## アーキテクチャ・リビジョン

アーキテクチャ・リビジョンによって、Nios II アーキテクチャの基本機能が強化され、すべての Nios II コアが影響を受けます。アーキテクチャの変更に伴い、新たなアーキテクチャの機能強化に対応するために、すべての Nios II コアに対するリビジョンが必要になります。例えば、アルテラが命令セットに新しい命令を追加した場合は、新しい命令が認識されるようにすべての Nios II コアを更新しなければなりません。表 6-2 に、Nios II アーキテクチャのリビジョンを示します。

バージョン	リリース月	説明
5.1	2005 年 10 月	変更なし。
5.0	2005 年 5 月	flushda 命令を追加。
1.1	2004 年 12 月	<ul style="list-style-type: none"> <li>● cpuid コントロール・レジスタを追加。</li> <li>● デバッグ・ツールで即値引数を使用できるように、break 命令の仕様を更新。</li> </ul>

表 6-2. Nios II アーキテクチャ・リビジョン ( 2 / 2 )

バージョン	リリース月	説明
1.01	2004 年 9 月	変更なし。
1.0	2004 年 5 月	Nios II プロセッサ・アーキテクチャの初回リリース。

## コア・ リビジョン

コア・リビジョンによって、既存の Nios II コアに変更が行われます。コア・リビジョンでは通常、確認されたバグの修正やアーキテクチャ・リビジョンのサポートを追加します。Nios II 開発キットのリリースごとに、すべての Nios II コアにリビジョンが行われるとは限りません。

### Nios II/f コア

表 6-3 に、Nios II/f コアのリビジョンを示します。

表 6-3. Nios II/f コア・リビジョン ( 1 / 2 )

バージョン	リリース月	説明
5.1	2005 年 10 月	変更なし。
5.0	2005 年 5 月	<ul style="list-style-type: none"> <li>● オプションの密結合メモリ・ポートを追加。設計者は 0 から 4 つの密結合インストラクション・マスタ・ポート、および 0 から 4 つの密結合マスタ・ポートを追加することができます。</li> <li>● データ・キャッシュ・ライン・サイズのコンフィギュレーションを可能にしました。設計者は、4、16、または 32 バイトのライン・サイズでデータ・キャッシュをコンフィギュレーションできます。従来は、データ・キャッシュのサイズは 4 バイトに固定されていました。</li> <li>● インストラクションおよびデータ・キャッシュのオプション（従来、キャッシュ・メモリは常に存在していました）。インストラクション・キャッシュが存在しない場合、Nios II コアはインストラクション・マスタ・ポートを持たず、密結合インストラクション・メモリを使用しなければなりません。</li> <li>● ハードウェアによって Cyclone II デバイス・サポートを確認。</li> <li>● HardCopy デバイスのフル・サポート（以前のバージョンでは、HardCopy デバイスをサポートするためにはワークアラウンドを実施しなくてはなりませんでした）。</li> </ul>

表 6-3. Nios II/f コア・リビジョン ( 2 / 2 )		
バージョン	リリース月	説明
1.1	2004 年 12 月	<ul style="list-style-type: none"> <li>● 乗算およびシフト操作に影響を与えるユーザ・コンフィギュレーションが可能なオプションを追加。設計者は以下の 3 つのオプションの 1 つを選択できるようになりました。               <ol style="list-style-type: none"> <li>(1) ターゲット・デバイス・ファミリで利用可能なエンベデッド乗算器リソースを使用 (従来からのオプション)。</li> <li>(2) ロジック・エレメントを使用して、乗算およびシフト・ハードウェアを実装 (新規オプション)。</li> <li>(3) ハードウェア乗算を省略。シフト操作は 1 ビットのシフトに 1 サイクルを要し、乗算演算はソフトウェアでエミュレートされます (新規オプション)。</li> </ol> </li> <li>● cpuid コントロール・レジスタを追加。</li> <li>● バグ修正：               <p>wrctl ienableによってディセーブルされた割り込みは、wrctl 命令の後の 1 クロック・サイクル間はイネーブルされたままでした。このような wrctl に続く命令が割り込みを受けることはなくなりました。(SPR 164828)</p> </li> </ul>
1.01	2004 年 9 月	<ul style="list-style-type: none"> <li>● ハードウェアによって Stratix II デバイス・サポートを確認。</li> <li>● バグ修正：               <ol style="list-style-type: none"> <li>(1) パイプライン内で、メモリへの格納の直後に同じメモリ位置からのロードが続き、そのメモリ位置が d キャッシュに保持されるときは、ロードが不正なデータを返すことがあります。この状況は、最適化をオフ (-O0) にして C コードをコンパイルすると発生する可能性があります。(SPR 158904)</li> <li>(2) SOPC Builder のトップ・レベルのシステム・モジュールには、非常に小さなアドレス空間をもつシステム用に余分で不要なポートが含まれていました。(SPR 155871)</li> </ol> </li> </ul>
1.0	2004 年 5 月	Nios II/f コアの初回リリース。

## Nios II/s コア

表 6-4 に、Nios II/s コアのリビジョンを示します。

表 6-4. Nios II/s コア・リビジョン		
バージョン	リリース月	説明
5.1	2005 年 10 月	変更なし。
5.0	2005 年 5 月	<ul style="list-style-type: none"> <li>● オプションの密結合メモリ・ポートを追加。設計者は 4 つの密結合インストラクション・マスタ・ポートにゼロを追加することができます。</li> <li>● インストラクション・キャッシュのオプション（従来、インストラクション・キャッシュは常に存在していました）。インストラクション・キャッシュが存在しない場合、Nios II コアはインストラクション・マスタ・ポートを持たず、密結合インストラクション・メモリを使用しなければなりません。</li> <li>● ハードウェアによって Cyclone II デバイス・サポートを確認。</li> <li>● HardCopy デバイスのフル・サポート（以前のバージョンでは、HardCopy デバイスをサポートするためにはワークアラウンドを実施しなくてはなりませんでした）。</li> </ul>
1.1	2004 年 12 月	<ul style="list-style-type: none"> <li>● 乗算およびシフト操作に影響を与えるユーザ・コンフィギュレーションが可能なオプションを追加。設計者は以下の 3 つのオプションの 1 つを選択できるようになりました。               <ol style="list-style-type: none"> <li>(1) ターゲット・デバイス・ファミリで利用可能なエンベデッド乗算器リソースを使用（従来からのオプション）。</li> <li>(2) ロジック・エレメントを使用して、乗算およびシフト・ハードウェアを実装（新規オプション）。</li> <li>(3) ハードウェア乗算を省略。シフト操作は 1 ビットのシフトに 1 サイクルを要し、乗算演算はソフトウェアでエミュレートされます（新規オプション）。</li> </ol> </li> <li>● 除算ハードウェアを ALU に含めるためのユーザ・コンフィギュレーション可能なオプションを追加。従来、このオプションは Nios II/f コアでしか利用できませんでした。</li> <li>● cpuid コントロール・レジスタを追加。</li> </ul>
1.01	2004 年 9 月	<ul style="list-style-type: none"> <li>● ハードウェアによって Stratix II デバイス・サポートを確認。</li> <li>● バグ修正： SOPC Builder のトップ・レベルのシステム・モジュールには、非常に小さなアドレス空間をもつシステム用に余分で不要なポートが含まれていました。(SPR 155871)</li> </ul>
1.0	2004 年 5 月	Nios II/s コアの初回リリース。

## Nios II/e コア

表 6-5 に、Nios II/e コアのリビジョンを示します。

バージョン	リリース月	説明
5.1	2005 年 10 月	変更なし。
5.0	2005 年 5 月	<ul style="list-style-type: none"> <li>ハードウェアによって Cyclone II デバイス・サポートを確認。</li> <li>HardCopy デバイスのフル・サポート (以前のバージョンでは、HardCopy デバイスをサポートするためにはワークアラウンドを実施しなくてはなりませんでした)。</li> </ul>
1.1	2004 年 12 月	cpuid コントロール・レジスタを追加。
1.01	2004 年 9 月	<ul style="list-style-type: none"> <li>ハードウェアによって Stratix II デバイス・サポートを確認。</li> <li>バグ修正: SOPC Builder のトップ・レベルのシステム・モジュールには、非常に小さなアドレス空間をもつシステム用に余分で不要なポートが含まれていました。(SPR 155871)</li> </ul>
1.0	2004 年 5 月	Nios II/e コアの初回リリース。

**JTAG デバッグ・モジュールのリビジョン** JTAG デバッグ・モジュールのリビジョンでは、Nios II プロセッサのデバッグ機能の強化、または JTAG デバッグ・モジュール・ロジック内で分離されたバグの修正が行われます。

表 6-6 に、JTAG デバッグ・モジュールのリビジョンを示します。

バージョン	リリース月	説明
5.1	2005 年 10 月	変更なし。
5.0	2005 年 5 月	HardCopy デバイスのフル・サポート (以前のバージョンでは、JTAG デバッグ・モジュールは HardCopy デバイスをサポートしていませんでした)。
1.1	2004 年 12 月	バグ修正: Nios II/s および Nios II/f コアを使用するときに、jmp、trap、または分岐命令にシーケンシャルに続く命令にハードウェア・ブレイクポイントを置いた場合は、誤ってトリガしていた可能性があります。(SPR 158805)

表 6-6. JTAG デバッグ・モジュールのリビジョン ( 2 / 2 )		
バージョン	リリース月	説明
1.01	2004 年 9 月	<ul style="list-style-type: none"> <li>● 機能拡張：               <ol style="list-style-type: none"> <li>(1) 命令アドレスに基づいてトリガする機能を追加。用途には、トレース・コントロールのトリガ（トレース・オン / オフ）、シーケンシャル・トリガ（下記参照）、およびトリガ・イン / アウト信号生成が含まれます。</li> <li>(2) トレース・バッファが満杯のときに、Nios II プロセッサを停止することなく収集を停止できるように、トレース収集を強化。</li> <li>(3) アーム付きトリガ-2 レベルのトリガ、つまり「アーム付きトリガ」をサポートして、「Event A then event B（イベント A の次にイベント B）」形式のトリガ定義に対応するようにトリガ・ロジックを強化。</li> </ol> </li> <li>● バグ修正：               <ol style="list-style-type: none"> <li>(1) Nios II/s コアでは、割り込み中に誤ったアドレスがトレース・データに記録されることがありました。(SPR 158033)</li> <li>(2) 特定の状況で、キャプチャされたトレース・データが望ましいトリガ位置も早くまたは遅れて開始するように表示されていました。(SPR 154467)</li> <li>(3) デバッグ中にハードウェア・ブレークポイントと割り込みが同時に発生すると、プロセッサがハングアップすることがありました。(SPR 154097)</li> </ol> </li> </ul>
1.0	2004 年 5 月	JTAG デバッグ・モジュールの初回リリース。

