

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。

S52014-2.2

はじめに

アルテラの最新エンハンスド・コンフィギュレーション・デバイスは、業界標準のフラッシュ・メモリと豊富な機能を備えたコンフィギュレーション・コントローラを組み合わせることにより、高集積コンフィギュレーション・ソリューションのニーズに応えます。シングル・チップ・コンフィギュレーション・ソリューションは、コンフィギュレーション時間を大幅に短縮するいくつかの新しい高度な機能を提供します。この章では、同時およびダイナミック・コンフィギュレーション、データ圧縮、クロック分周、外部フラッシュ・メモリ・インタフェースなどのエンハンスド・コンフィギュレーション・デバイス機能のハードウェアおよびソフトウェア実装について説明します。エンハンスド・コンフィギュレーション・デバイスには、EPC4、EPC8、およびEPC16 デバイスが含まれます。

同時コンフィ ギュレーション

コンフィギュレーション・データは、エンハンスド・コンフィギュレーション・デバイスから DATA ライン上の SRAM ベースのデバイスに送信されます。DATA ラインは、エンハンスド・コンフィギュレーション・デバイス上の出力であり、SRAM ベース・デバイスへの入力です。

これらの DATA ラインは、アルテラ Quartus® II ソフトウェアの **Convert Programming Files** ウィンドウの *Bit n* ラインに対応しています。例えば、SRAM オブジェクト・ファイル (.sof) を指定して Quartus II ソフトウェアの *Bit0* を使用する場合、.sof はエンハンスド・コンフィギュレーション・デバイスから SRAM ベース・デバイスへの DATA[0] ライン上を送信されます。

エンハンスド・コンフィギュレーション・デバイスは、サポートされる各種コンフィギュレーション手法により、多数のデバイスを同時にコンフィギュレーションできます。

サポートされる手法およびガイドライン

エンハンスド・コンフィギュレーション・デバイスを使用して、アルテラの SRAM ベース・プログラマブル・ロジック・デバイス (PLD) をコンフィギュレーションする方法がいくつかあります。

- 1 ビット・パッシブ・シリアル (PS)
- 2 ビット PS
- 4 ビット PS
- 8 ビット PS

■ ファースト・パッシブ・パラレル (FPP)

さらに、洗練されたコンフィギュレーション・セットアップを実現するために、これらのコンフィギュレーション手法をダイナミック・コンフィギュレーション・オプション（以前はページ・モード操作と呼ばれていました）と併用することができます。

FPP コンフィギュレーション・モードでは、エンハンスド・コンフィギュレーション・デバイスからの 8 本の DATA [7..0] ラインを使用しますが、これは Stratix® シリーズおよび APEX™ II デバイスのコンフィギュレーションに使用できます。コンフィギュレーション時間を短縮するために、FPP コンフィギュレーションはターゲット・デバイスに、1 クロック・サイクルあたり 8 ビットのコンフィギュレーション・データを提供します。



コンフィギュレーション手法について詳しくは、「エンハンスド・コンフィギュレーション・デバイス・データシート」、「Application Note 116: Configuring SRAM-Based LUT Devices」、または「Configuring Stratix & Stratix GX Devices」を参照してください。




アルテラはエンハンスド・コンフィギュレーション・デバイスの供給元の追加を発表しました。詳しくは、プロセス変更通知 PCN0506: *Addition of Intel Flash Memory As Source For EPC4, EPC8 & EPC16 Enhanced Configuration Devices* およびホワイトペーパー「Using Intel Flash Memory Based EPC4, EPC8, and EPC16」を参照してください。

***n* ビット PS モードを使用した同時コンフィギュレーション**

n ビット ($n=1, 2, 4, \text{および} 8$) PS コンフィギュレーション・モードでは、エンハンスド・コンフィギュレーション・デバイスが同時に SRAM ベース・デバイスまたはデバイス・チェーンをコンフィギュレーションすることができます。さらに、これらのデバイスは同じデバイス・ファミリや集積度である必要はなく、アルテラの SRAM ベース・デバイスの任意の組み合わせが可能です。個々のエンハンスド・コンフィギュレーション・デバイスの DATA ラインを各ターゲット・デバイスに使用できます。各 DATA ラインは、デバイスのデジター・チェーンにも供給できます。

Quartus II ソフトウェアでは、*n* ビット PS のコンフィギュレーション・モードの選択のみ可能です。ただし、これらのモードを使用して 1～8 個の任意のデバイス数をコンフィギュレーションできます。*n* ビット PS モードを使用して SRAM ベース・デバイスをコンフィギュレーションするときは、コンフィギュレーション時間が最短になるように、表 3-1 を使用して、適切なコンフィギュレーション・モードを選択します。

 3-20ページの「コンフィギュレーション・スペース・サイズの計算」で説明するように、モードの選択が使用するメモリ容量に影響を与えます。

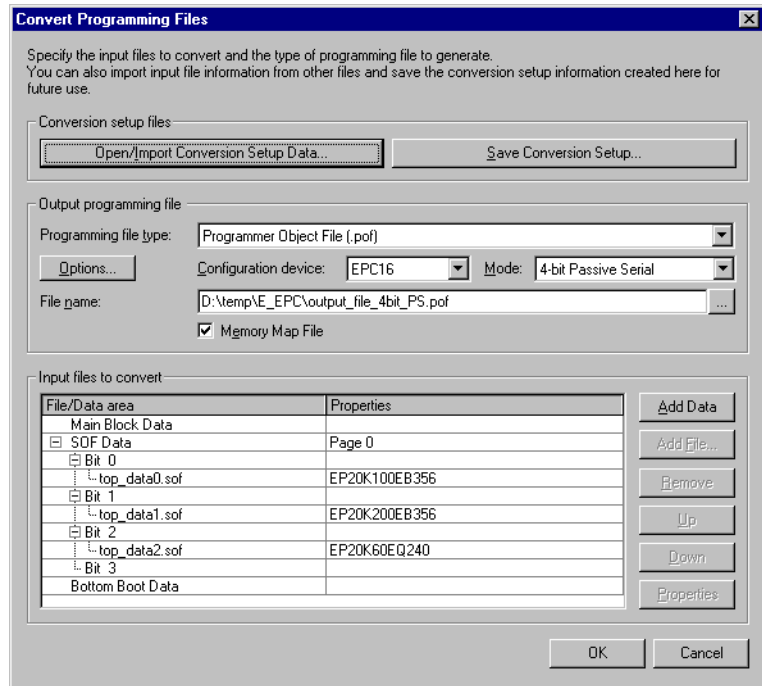
デバイス数 (1)	推奨コンフィギュレーション・モード
1	1ビット PS
2	2ビット PS
3	4ビット PS
4	4ビット PS
5	8ビット PS
6	8ビット PS
7	8ビット PS
8	8ビット PS

表 3-1 の注：

- (1) 各 DATA ラインがデバイスのデジター・チェーンでなく、1 個のデバイスのみをコンフィギュレーションすると仮定します。

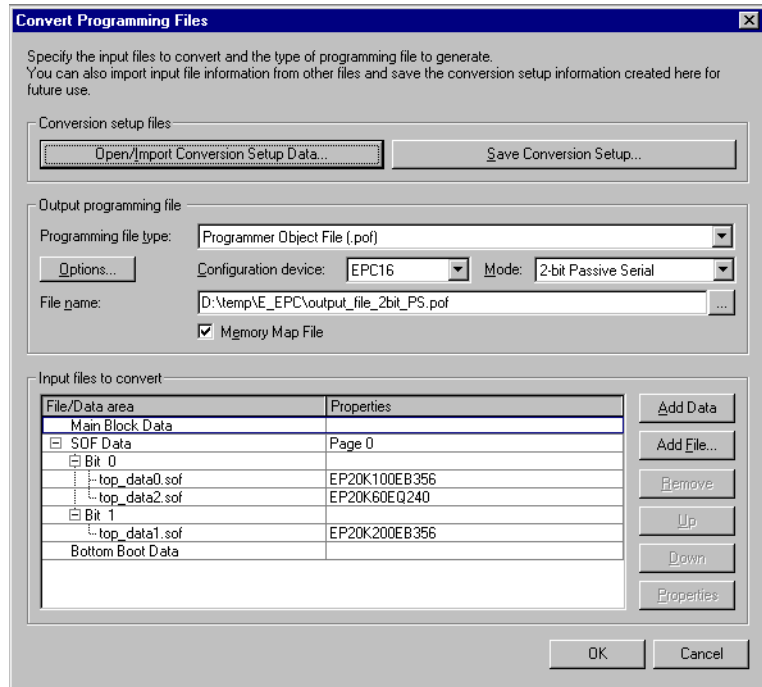
例えば、3 個の SRAM ベース・デバイスをコンフィギュレーションする場合は、4 ビット PS モードを使用します。DATA0、DATA1、および DATA2 ラインの場合、対応する .sof データはコンフィギュレーション・デバイスから SRAM ベース PLD に送信されます。DATA3 の場合、Quartus II ソフトウェアの対応する Bit3 ラインをブランクにしておくことができます。プリント基板 (PCB) 上では、エンハンスド・コンフィギュレーション・デバイスからの DATA3 ラインを接続しないでおきます。図 3-1 に、この手法のための **Quartus II Convert Programming Files** ウィンドウ (Tools メニュー) のセットアップを示します。

図 3-1. nビット PS モードを使用したデバイスの
コンフィギュレーションのためのソフトウェア設定



あるいは、2個のSRAMベース・デバイスを1本のDATAラインにデイズ・チェーンし、他のDATAラインではそれぞれ1個のデバイスをドライブすることができます。例えば、2ビットPSモードを使用して、DATA Bit0 (EP20K100EおよびEP20K60Eデバイス)で2個のSRAMベース・デバイスをドライブし、DATA Bit1で3番目のデバイス (EP20K200Eデバイス)をドライブすることができます。この2ビットPSコンフィギュレーション手法では、コンフィギュレーション・フラッシュ・メモリに必要なスペースは減少しますが、合計システム・コンフィギュレーション時間 (図3-2)は長くなる場合があります。

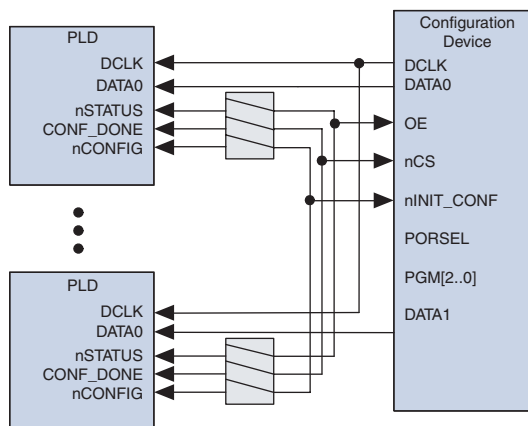
図 3-2. 2 個の SRAM ベース・デバイスの 1 本の DATA ラインへの
デジター・チェイン



デザイン・ガイドライン

デバッグのために、アルテラでは各 PLD とコンフィギュレーション・デバイスの間で、nSTATUS、nCONFIG、CONF_DONE などのコントロール・ラインを分離することを推奨しています。1 個のスイッチを使用して、エンハンスド・コンフィギュレーション・デバイスにフィードバックされるコントロール信号を管理することにより、コントロール・ラインを分離できます。図 3-3 に、エンハンスド・コンフィギュレーション・デバイスとターゲットの PLD 間の接続例を示します。

図 3-3. コントロール・ラインへのデバッグ用スイッチの使用例




ダイナミック・ コンフィギュ レーション (ページ・モード) の実装について

エンハンスド・コンフィギュレーション・デバイスでのページによって、全体で1個または複数のアルテラ PLD を使用するシステムに対して、さまざまなコンフィギュレーションを編成したり、格納することができます。このダイナミック・コンフィギュレーション (またはページ・モード) 機能により、システムは使用している PLD を異なるコンフィギュレーション・ファイルでダイナミックにリコンフィギュレーションすることができます。

別のページを使用して、異なる規格 (例えば、標準 I/O 規格やメモリなど) をサポートするコンフィギュレーション・ファイルを格納できます。あるいは、別のページでシステムを異なるモードに置くことができます。例えば、ページ 0 にはデータ・パケットのみを処理する PLD 用のコンフィギュレーション・ファイル (.sof) を、ページ 1 にはデータ・パケットと音声パケットを処理する同じ PLD 用のコンフィギュレーション・ファイルを含めることができます。

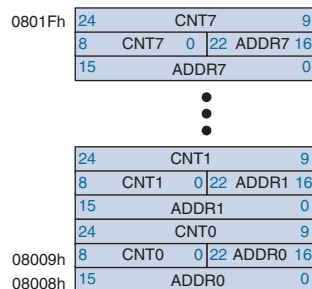
ページをダイナミックに切り替えることができるため、コンフィギュレーション・デバイスを再プログラムする必要なく、デバッグ用の各種リビジョンでアルテラ・デバイスをコンフィギュレーションすることも可能です。例えば、「スタンバイ状態」のデバイスを別の機能を実行するようにコンフィギュレーションしてから、それを元のコンフィギュレーション・ファイルでリコンフィギュレーションして元に戻すこともできます。

ページとは、システム内のすべてのPLD用のコンフィギュレーション・データを保持するフラッシュ・メモリ・スペース内の1つのセクションのことです。システム内のPLDの個数に関係なく、1ページに1つのシステム・コンフィギュレーションが格納されます。各ページのサイズはダイナミックで、エンハンスド・コンフィギュレーション・デバイスが再プログラムされるたびに变化する可能性があります。エンハンスド・コンフィギュレーション・デバイスは、最大8ページのコンフィギュレーション・データ、または8つのシステム・コンフィギュレーションをサポートします。ページ数もコンフィギュレーション・デバイスの集積度に制限されます。

 システムに必要なページ数は、システム内のPLDの個数には関係ありませんが、固有のシステム・コンフィギュレーション数に依存します。

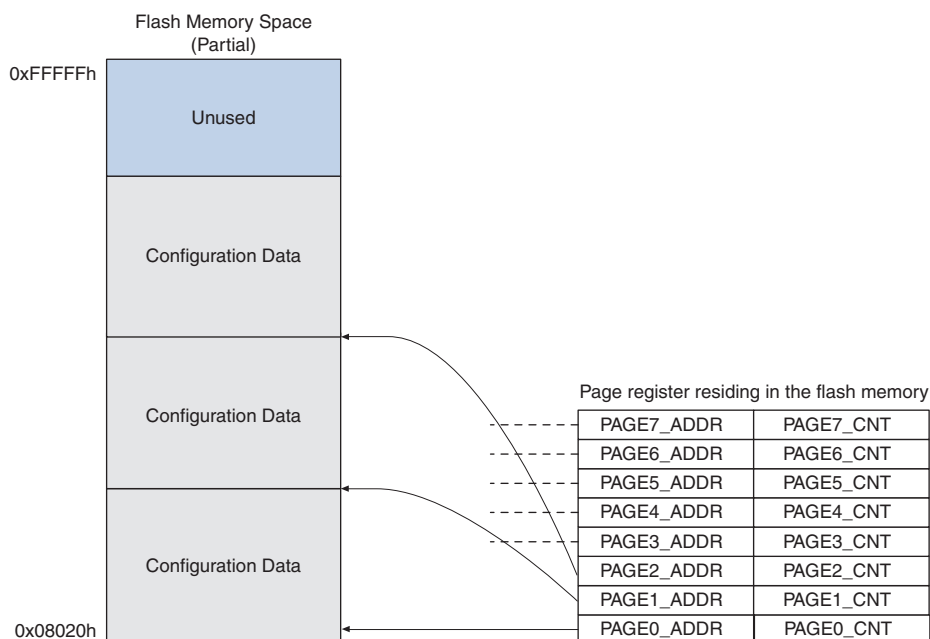
外部ページ・モードの入力ピン PGM[2..0] は、PLDのコンフィギュレーション中に、どのページを使用するかを決定し、ページ・ポインタはデータの位置を決定します。各ページ・ポインタは、開始アドレス・レジスタとレンジス・カウント・レジスタで構成されます。ワード・アドレス指定可能な開始アドレス・レジスタ（23ビット）は、フラッシュ・メモリ内のページ開始位置を決定するのに使用されます。カウント・レジスタ（25ビット）は、ニブル（1バイトの半分に対応する4ビットのグループ）でカウントされるページの長さを決定します。図 3-4 に、オプション・ビット・スペースとそのアドレス位置のブロック図を示します。

図 3-4. オプション・ビット・メモリ・マップ




例えば、EPC16 コンフィギュレーション・デバイスのページは、ワード・アドレス 0x08020h から 0xFFFFFh までの間で開始しなければならない、他のページとオーバーラップすることはできません。3 ページを使用した EPC16 ページ・モードの例がありますので、図 3-5 を参照してください。

図 3-5. EPC16 ページ・モードの実装例



コンフィギュレーション中に、PGM[2..0] ピンによって異なるページが選択されます。これらのピンは、8 ページ（または 8 つのシステム・コンフィギュレーション）から 1 ページを選択するのに使用されます。PGM[2..0] ピンは、コンフィギュレーション・データがターゲット PLD に送信される前に一度サンプリングされます。

PGM[2..0] ピンを、不適当なページ（存在しないページやブランク・ページなど）を選択するように設定すると、エンハンスド・コンフィギュレーション・デバイスは不正な状態になります。この状態から復帰させるには、有効なページを選択するように PGM[2..0] ピンを設定してから、ボードの電源を入れ直す以外に方法はありません。

 適切なコンフィギュレーションを実行するには、PGM[2..0] ピンが有効なページを選択するように設定するだけです。

各ページ内で、システムに必要な数のコンフィギュレーション・ファイルを格納できます。フラッシュ・メモリのサイズで決まる物理的な制限（例：EPC16 デバイスの場合は 0xFFFFFh）を除いて、ページの長さには制限はありません。ただし、すべてのページは連続している必要があります。

ソフトウェアの実装（プログラミング・ファイルの変換）

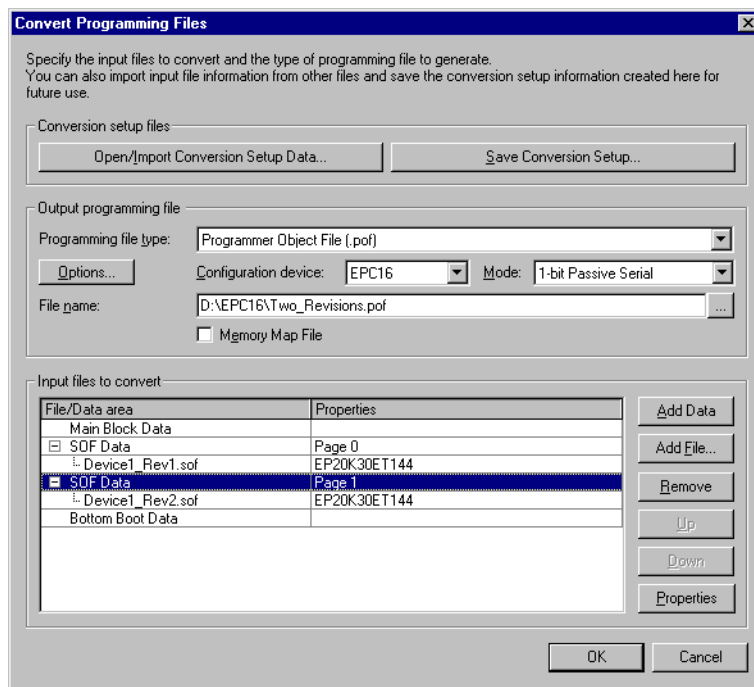
Quartus II ソフトウェアの **Convert Programming Files** ウィンドウ（Tools メニュー）により、エンハンスド・コンフィギュレーション・デバイスのプログラマ・オブジェクト・ファイル（.pof）を作成し、ダイナミック・コンフィギュレーション機能をイネーブルにすることができます。



パッシブ・パラレル非同期 (PPA) およびパッシブ・パラレル同期 (PPS) コンフィギュレーション・モードは、エンハンスド・コンフィギュレーション・デバイスではサポートされていません。これらのモードのいずれかを選択した場合、Quartus II ソフトウェアはエンハンスド・コンフィギュレーション・デバイスの .pof が生成されたときにエラー・メッセージをレポートします。

Convert Programming Files ウィンドウには、**Input files to convert** ダイアログに、**SOF Data** エントリ (.sof) があります。各 **SOF Data** エントリは、個別のシステム・コンフィギュレーションを参照します。図 3-6 に、1 個の APEX デバイスを持ち、ページ 0 と 1 の 2 ページを使用するシステムのセットアップを示します。この 2 ページそれぞれに、同じ APEX デバイス用で異なるバージョンのコンフィギュレーション・ファイルがあります。

図 3-6. ページ・モード例の使用



特定のページまたは **SOF Data** エントリをポイントするページ・ポイントを設定するには、**SOF Data** を選択して、**Properties** をクリックします。**Properties** をクリックして、**SOF Data Properties** ウィンドウを開いて、そのウィンドウで選択された **SOF Data** をポイントするページ・ポイントを選択できます。**SOF Data Properties** ウィンドウを使用して変更を行わない場合、デフォルト・ページは 0 です。コンフィギュレーション・デバイスの各 **SOF Data** エントリは、固有のページ番号を持っていなければなりません。図 3-7 に、(図 3-6 からの) **Device1_Rev2.sof** を含む **SOF Data** セクションに割り当てられたページ・ポイントを示します。

図 3-7. ページ選択のためのソフトウェア設定

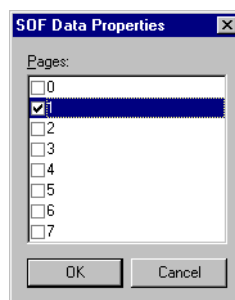
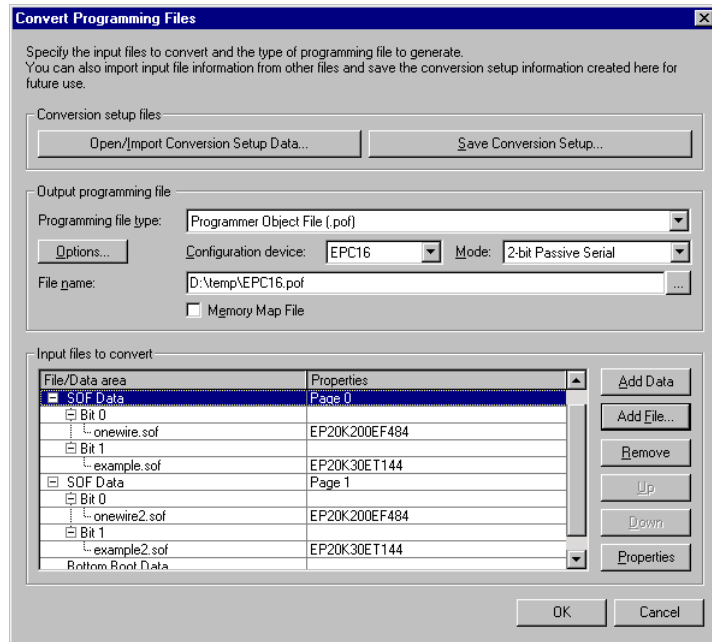


図 3-8 に、2 ビット PS コンフィギュレーション・モードを使用して、各デザインの 2 つのリビジョンを格納する複数ページで、2 つの異なる APEX デバイスを同時にコンフィギュレーションするより複雑なセットアップを示します。システム全体の 2 つのコンフィギュレーションには、4 つのコンフィギュレーション・ファイル（デバイス数 x 固有のシステム・コンフィギュレーション数）が必要です。

図 3-8. 2つのシステム・コンフィギュレーションによる2個のデバイスの同時コンフィギュレーション



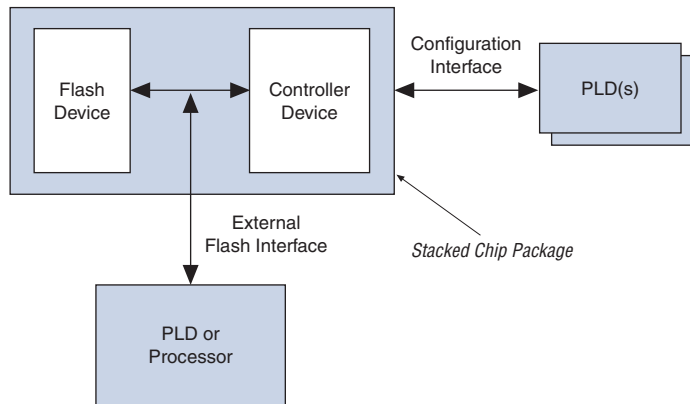
Memory Map File オプションを選択することにより、フラッシュ・メモリのアドレス位置を記述する **Quartus II** メモリ・マップ出力ファイル (**.map**) が生成されます。この情報は、一般に外部フラッシュ・インタフェース機能を使用する場合に役立ちます。

外部フラッシュ・メモリ・インタフェース

エンハンスド・コンフィギュレーション・デバイスは外部フラッシュ・インタフェースをサポートしており、それによりコントローラ外部のデバイスはエンハンスド・コンフィギュレーション・デバイスのフラッシュ・メモリにアクセスできます。フラッシュ・メモリは、ブートまたはプロセッサ用アプリケーション・コードを格納するのに使用するか、あるいはプロセッサおよび PLD 用の汎用メモリとして使用できます。

図 3-9 に、エンハンスド・コンフィギュレーション・デバイスで使用可能なインタフェースを示します。

図 3-9. エンハンスド・コンフィギュレーション・デバイスの
インタフェース



オン・ボード・プログラマブル・ロジック (Stratix シリーズ・デバイス) のリモート・アップデート機能を必要とするアプリケーション、およびソフト・エンベデッド・プロセッサ・コア (Nios® エンベデッド・プロセッサなど) を使用するアプリケーションは、一般に外部フラッシュ・メモリ・インタフェース機能を使用します。

ソフト・エンベデッド・プロセッサ・コア・アプリケーションでは、コントローラはフラッシュ・メモリに格納されているコンフィギュレーション・データを使用して、プログラマブル・ロジックをコンフィギュレーションします。正常なコンフィギュレーションでは、エンベデッド・プロセッサは外部フラッシュ・インタフェースを使用して、同じフラッシュ・メモリからコードをブートアップして実行するため、スタンドアロンのフラッシュ・メモリ・デバイスは必要ありません。

リモート・システム・コンフィギュレーション機能を必要とするアプリケーションでは、プロセッサまたはPLDは外部フラッシュ・インタフェースを使用してアップデートされたコンフィギュレーション・イメージをフラッシュ・メモリ内の新しいページに格納することができます (外部フラッシュ・インタフェースとダイナミック・コンフィギュレーションの併用)。新しいコンフィギュレーション・データはローカル・インテリジェント・ホストから、あるいはインターネットを通じて取得できます。システムを新しいページでリコンフィギュレーションすると、システム・コンフィギュレーションがアップデートされます。



エンハンスド・コンフィギュレーション・デバイスによるリモートおよびローカル・システム・アップデートの実装については、「Stratix ハンドブック」の「Stratix & Stratix GX デバイスのリモート・システム・コンフィギュレーション」または「Stratix II ハンドブック」の「Stratix II & Stratix II GX デバイスのリモート・システム・アップグレード」の章を参照してください。

現在、EPC4 および EPC16 コンフィギュレーション・デバイスは外部フラッシュ・インタフェースをサポートしています。他のエンハンスド・コンフィギュレーション・デバイスでのこの機能のサポートについてはお問い合わせください。

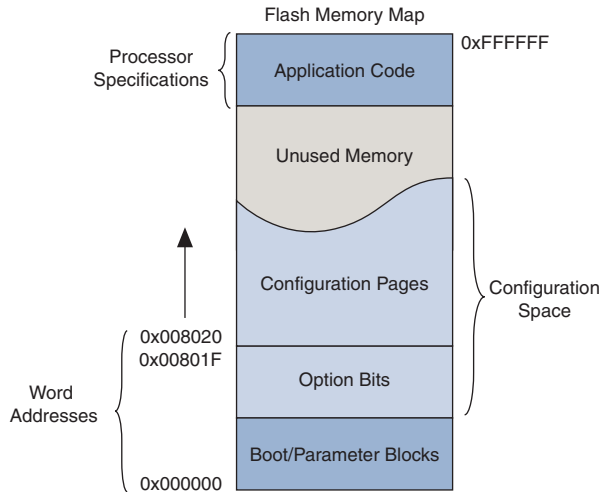
フラッシュ・メモリ・マップ

エンハンスド・コンフィギュレーション・デバイスのフラッシュ・メモリは、ロジカル（コンフィギュレーションおよびプロセッサ・スペース）とフィジカル（フラッシュ・データ・ブロック境界）の2つのカテゴリに分類することができます。コンフィギュレーション・スペースは、コンフィギュレーション・オプション・ビットとコンフィギュレーション・データを格納するのに使用されるメモリ領域で構成されます。また、プロセッサ・スペースは、ブートおよびアプリケーション・コードを格納するのに使用されるメモリ領域で構成されます。

ロジカル・ディビジョン

すべてのエンハンスド・コンフィギュレーション・デバイスで、コンフィギュレーション・オプション・ビットは、ワード・アドレス 0x008000 から 0x00801F（バイト・アドレス 0x010000 から 0x01003F）に格納されます。これらのビットは、コンフィギュレーション・モードの選択、圧縮モードの選択、クロック・ディバイダの選択など、さまざまなコントローラ機能をイネーブルするために使用されます。すべてのエンハンスド・コンフィギュレーション・デバイスで、コンフィギュレーション・データはワード・アドレス 0x008020 またはバイト・アドレス 0x010040 から始まる位置に格納されます。コンフィギュレーション・スペースの終了アドレスは固定されておらず、エンハンスド・コンフィギュレーション・デバイスを使用してコンフィギュレーションされた PLD の個数と集積度、およびページ数によって決まります。コンフィギュレーション・スペースより上位の残りのすべてのアドレス位置は、プロセッサ・アプリケーション・コードに使用可能です。ブート・スペースの範囲は、アドレス 0x000000 から 0x007FFF までです。ブートおよびアプリケーション・コードのスペースは、いずれも外部プロセッサまたは PLD で使用することを目的としています。図 3-10 に、EPC16 デバイス内部のフラッシュ・メモリ・マップを示します。

図 3-10. EPC16 フラッシュ・メモリ・マップ



フィジカル・ディビジョン

逆に、フィジカル・ディビジョンは個々に書き込みおよび消去可能なフラッシュ・データ・ブロックです。例えば、Sharp フラッシュ・ベースの EPC16 デバイスには、2つのブート・ブロック、6つのパラメータ・ブロック、および31のメイン・データ・ブロックに分割された16 Mビットの Sharp フラッシュ・メモリが含まれています。これらのフィジカル・ディビジョンはフラッシュ・メモリまたはベンダごとに異なるため、外部フラッシュ・インタフェースを使用してフラッシュ・メモリを消去または書き込む場合は考慮する必要があります。最初のプログラミング後に、このインタフェースを読み出し専用インタフェースとして使用する場合には、これらの区分は重要ではありません。



エンハンスド・コンフィギュレーション・デバイスのフラッシュ・メモリについて詳しくは、下記のドキュメントを参照してください。

- Micron フラッシュ・ベースの EPC4 については、www.micron.com の *Micron Flash Memory MT28F400B3 Data Sheet* を参照してください。
- Sharp フラッシュ・ベースの EPC16 については、www.sharpsma.com の *Sharp LHF16J06 Data Sheet Flash Memory Used in EPC16 Devices* を参照してください。


- *Intel Advanced Boot Block Flash Memory (B3) 28F008/800B3, 28F016/160B3, 28F320B3, 28F640B3 Datasheet* については、www.intel.com を参照してください。

インタフェースの可用性と接続

フラッシュ・メモリ・ポートは、内部コントローラと外部デバイス間で共有されます。プロセッサまたはPLDは、コントローラがインタフェースを使用していないときにのみ、外部フラッシュ・インタフェースを使用してフラッシュ・メモリにアクセスできます。したがって、内部コントローラはバスのプライマリ・マスタで、外部デバイスはセカンダリ・マスタです。

フラッシュ・メモリ・ポート（アドレス、データ、およびコントロール）は、内部でコントローラ・デバイスに接続されています。さらに、これらのポートはパッケージのピンに接続され、外部インタフェースを提供します。エンハンスド・コンフィギュレーション・デバイスのイン・システム・プログラミング中、およびPLDのコンフィギュレーション中には、コントローラはフラッシュ・メモリへの内部インタフェースを使用し、外部インタフェースは使用不可にします。外部デバイスは、イン・システムプログラミングおよびコンフィギュレーションの全期間中は競合を防止するために、すべての接続（アドレス、データ、およびコントロール）をトライ・ステートする必要があります。

イン・システム・プログラミングおよびコンフィギュレーションの完了時、内部コントローラはフラッシュ・メモリへのインタフェースをトライ・ステートにし、アドレス・ラインおよびコントロール・ラインのウィーク内部プルアップ抵抗をイネーブルにし、さらにデータ・ライン上のバス・ホールド回路をイネーブルにします。ここで、内部フラッシュ・インタフェースがディセーブルされ、外部フラッシュ・インタフェースが使用可能になります。


 外部フラッシュ・インタフェース機能を使用しない場合、競合を回避するためにボード上でフラッシュに関連する大部分のピンは未接続のままにしておくことが必要です。データシートおよびピン配置表に概要が示されているこのガイドラインには、いくつかの例外があります。



詳細な回路図については、「エンハンスド・コンフィギュレーション・デバイス・データシート」を参照してください。


Quartus II ソフトウェア・サポート

Convert Programming Files ウィンドウを使用して、フラッシュ・メモリのプログラミング・ファイルを生成できます。JTAG (Joint Test Action Group) を使用するか、外部フラッシュ・インタフェースを介して、イン・システムでフラッシュ・メモリをプログラムすることができます。フラッシュ・メモリをイン・システムでプログラムする場合は、**.pof** を選択してください。この **.pof** は、Jam™ STAPL (Standard Test and Programming Language) ファイル (**.jam**) またはイン・システム・プログラミング用の Jam バイト・コード・ファイル (**.jbc**) に変換することもできます。外部フラッシュ・インタフェースを介してフラッシュ・メモリをプログラムすれば、このウィンドウから **.hexout** を作成できます。

 エンハンスド・コンフィギュレーション・デバイスのプログラミングに使用される **.hexout** は、SRAM PLD 用に生成される **.hexout** コンフィギュレーション・ファイルとは異なります。

PLD コンフィギュレーション・ファイルに加えて、**Convert Programming Files** ウィンドウを使用してフラッシュ・メモリにプロセッサ・ブートおよびアプリケーション・コードをプログラムすることができます。ブート・コードを含む **.hex** ファイルをウィンドウの **Bottom Boot Data** セクションに追加できます。同様に、アプリケーション・コードを含む **.hex** ファイルを **Main Block Data** セクションに追加できます。これらのファイルは、相対または絶対アドレッシングを使用してフラッシュ・メモリに格納することができます。アドレッシングのタイプを選択するには、**Bottom Boot Data** または **Main Block Data** セクションをハイライトし、**Properties (Convert Programming Files** ウィンドウ) をクリックします。

相対アドレッシング・モードにより、Quartus II ソフトウェアはメモリ内のファイルの位置を見つけることができます例えば、Quartus II ソフトウェアは、常にアドレス位置 0x000000 で始まるブート・コードを格納します。このデータは上位アドレスに向かって増加します。

 EPC16 コンフィギュレーション・デバイスの最大ブート・ファイル・サイズは、32 K ワードまたは 64 K バイトです。ブート・コードはブートおよびフラッシュ・メモリ・パラメータ・ブロックに制限されています。

Main Block Data に相対アドレッシング・モードを選択すると、Quartus II ソフトウェアは情報の最後のバイトを最上位アドレス (例えば、0x1FFFFFF) に揃えます。したがって、開始アドレスは **.hex** ファイルのサイズによって決まります。この項で説明した **.map** ファイルを使用すれば、簡単にアプリケーション・コードの開始アドレスを取得できます。

逆に、絶対アドレッシング・モードでは、Quartus II ソフトウェアはブートまたはアプリケーションの .hex ファイルのデータを .hex ファイル内に指定されたアドレス位置に格納します。このモードが選択されているときは、正しいオフセットの .hex ファイルを作成して、コンフィギュレーション・データの格納に使用されるアドレスとオーバラップしていないことを確認します。

図 3-11 に、エンハンスト・コンフィギュレーション・デバイス用 .pof および .map ファイルを作成するための **Convert Programming Files** ウィンドウ・セットアップのスクリーン・ショットを示します。


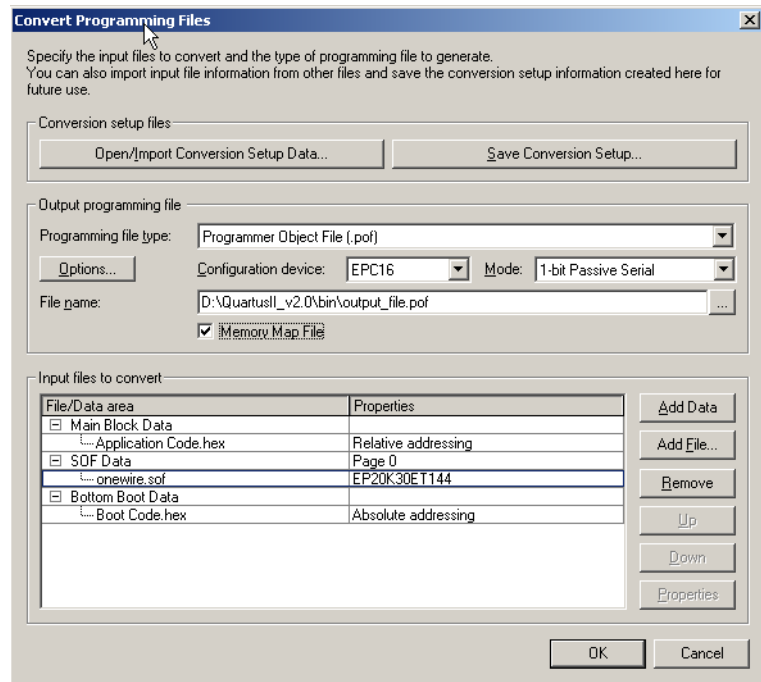
 このウィンドウの **Bottom Boot Data** および **Main Block Data** セクションに追加できるのは、1つの .hex ファイルのみです。

図 3-11. フラッシュ・メモリへのブートおよびアプリケーション・コードの格納



Quartus II Convert Programming Files ウィンドウを使用して、外部フラッシュ・インタフェース機能に特有の **.hexout** ファイルと **.map** ファイルの2つのファイルを作成することができます。**.hexout** にはフラッシュ・メモリのイメージが含まれ、**.map** ファイルにはメモリ・マップ情報が含まれます。**.hexout** は、外部プロセッサまたは PLD が外部フラッシュ・インタフェースを介してフラッシュ・メモリをプログラムするために使用できます。**.map** ファイルには、ブート・コードの開始アドレスと終了アドレス、コンフィギュレーション・ページのデータ、およびアプリケーション・コードが含まれます。


.hexout を使用して、ブランク状態のエンハンスド・コンフィギュレーション・デバイスおよび/またはフラッシュ・メモリのアップデート部分（例えば、新しいコンフィギュレーション・ページ）をプログラムできます。このファイルは、内部 16 進ファイル・フォーマットを使用し、16 M ビット、または 2 M バイトのデータを保持します。表 3-2 に **.map** ファイルのフォーマットを示します。

ブロック	開始アドレス	終了アドレス
BOTTOM BOOT	0x00000000	0x0000001F
オプション・ビット	0x00010000	0x0001003F
ページ 0	0x00010040	0x0001AD7F
メイン	0x001FFFE0	0x001FFFFFFF

表 3-2 の注：

(1) このファイル内のアドレスはすべてバイト・アドレスです。

部分フラッシュ・メモリ・アップデートを実行するには、**.map** ファイルに提供されているメモリ・マップ情報を使用して **.hexout** の関連部分を選択します。

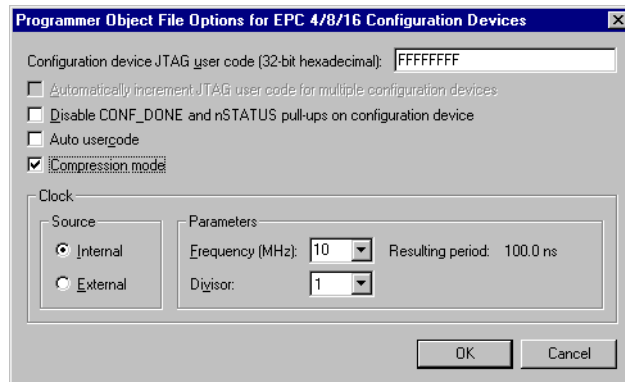
 コンフィギュレーション・データおよびプロセッサ・スペース・データは、同じフィジカル・データ・ブロック内に存在する可能性があります。このような場合、フィジカル・データ・ブロックを消去すると、コンフィギュレーション・データとプロセッサ・データの両方に影響するため、両方のアップデートが必要になります。この状況は、次に利用可能な完全データ・ブロックからアプリケーション・データを格納することによって回避できます。

データ圧縮

エンハンスド・コンフィギュレーション・デバイスは、コンフィギュレーション・データを標準的なデザインで 1.9× だけ圧縮する効率的な圧縮アルゴリズムをサポートしており、デバイスのサイズを実質的に倍増します。エンハンスド・コンフィギュレーション・デバイスに対する正しい集積度を選択するには、非圧縮のコンフィギュレーション・スペースの合計サイズを事前に計算する必要があります。

図 3-12 に示すように、**Options (Convert Programming Files ウィンドウ)** をクリックすると、プログラミング・ファイル・タイプとして選択された **po** により、**Programming Object File Options** ウィンドウの **Compression mode** オプションをオンにすることができます。

図 3-12. 圧縮モードの選択



コンフィギュレーション・スペース・サイズの計算

1 ビット PS コンフィギュレーション・モードを使用して、複数のデバイスをシリアルにコンフィギュレーションするとき、同じ DATA ラインを通してすべてのコンフィギュレーション・データが送信され、デバイスは共にダイジェスト・チェーンされます。したがって、非圧縮コンフィギュレーション・データの合計サイズは、SRAM ベース・デバイスのコンフィギュレーション・ファイル・サイズに使用するページ数を乗算した値の合計になります。

n ビット PS コンフィギュレーション・モードを使用して複数のデバイスを同時にコンフィギュレーションするとき、各 SRAM ベース・デバイスにはエンハンスド・コンフィギュレーション・デバイスからの専用の DATA ラインがあります。非圧縮コンフィギュレーション・スペースの合計サイズは、最大デバイスのコンフィギュレーション・ファイル・サイズに n ($n=1, 2, 4$, または 8) を乗算し、さらに使用ページ数を乗算した値になります。例えば、4 ビット PS コンフィギュレーション・モードを使用して 3 つのデバイスを同時にコンフィギュレーションする場合、非圧縮コンフィギュレーション・スペースの合計サイズは、最大デバイスのコンフィギュレーション・ファイル・サイズの 4 倍になります。

FPP コンフィギュレーション・モードでは、非圧縮コンフィギュレーション・スペースの合計サイズは SRAM ベース・デバイスのコンフィギュレーション・ファイル・サイズに使用するページ数を乗算した値になります。

SRAM ベース・デバイスのコンフィギュレーション・ファイル・サイズについては、「コンフィギュレーション・ハンドブック」を参照してください。

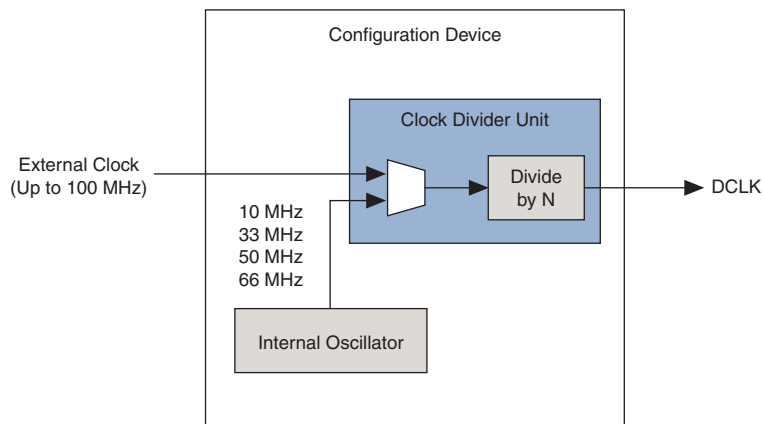
クロック・ ディバイダ

クロック・ディバイダの値は、DCLK 周波数を求めるのに使用するクロック周波数の除数、またはデータが SRAM ベース・デバイスにクロック・インされる速度を指定します。クロック入力およびディバイダの設定を選択している間は、ターゲットとなる SRAM デバイス・ファミリの最大 DCLK 入力周波数を考慮する必要があります。SRAM ベース・デバイスの DCLK タイミング仕様については、「コンフィギュレーション・ハンドブック」を参照してください。

設定およびガイドライン

図 3-13 に示すとおり、エンハンスド・コンフィギュレーション・デバイスは内部オシレータまたは外部クロック・ソースのいずれかを使用して、データを SRAM ベース・デバイスにクロックできます。エンハンスド・コンフィギュレーション・デバイスの内部オシレータは、10 MHz、33 MHz、50 MHz、または 66 MHz の公称速度で動作します。最小速度および最大速度は、「エンハンスド・コンフィギュレーション・デバイス・データシート」に示します。さらに、エンハンスド・コンフィギュレーション・デバイスは、最大 100 MHz の速度で動作する外部クロック・ソースを受け入れることができます。

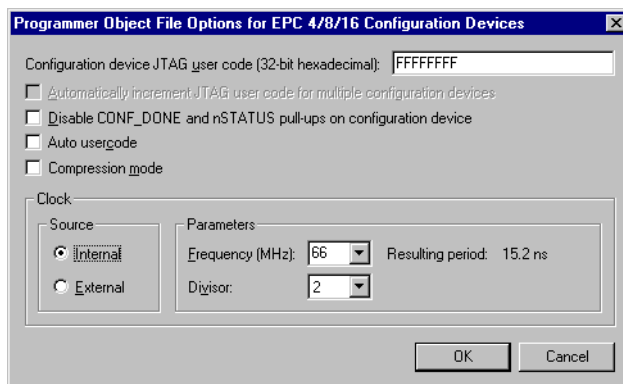
図 3-13. エンハンスド・コンフィギュレーション・デバイスのクロック・ディバイダ・ユニット



ソフトウェアの実装

図 3-14 に示すように、プログラミング・ファイル・タイプとして **pof** を選択 (Convert Programming Files ウィンドウ) した状態で、**Programming Object File Options** ウィンドウでクロック・ソースとクロック速度を選択できます。**Frequency (MHz)** ドロップ・ダウン・メニューで、適切な外部クロック周波数を入力し、クロック・ソースの設定に関係なく、除数のリストから任意の値を選択できます。

図 3-14. クロック・ソースおよびクロック除数を設定するソフトウェア



まとめ

エンハンスド・コンフィギュレーション・デバイスは、より柔軟なコンフィギュレーションおよび迅速な「Time-to-Market」を提供するシングル・チップ・コンフィギュレーション・ソリューションです。データ圧縮、複数のクロック・ソース、クロック分周、およびパラレルまたは同時プログラミングなどの機能により、コンフィギュレーション時間が大幅に短縮されるとともに、ダイナミック・コンフィギュレーション・モードや外部フラッシュ・インタフェースによって、より高いレベルのインテリジェント・システム・コンフィギュレーションを実現します。

