

はじめに

このアプリケーション・ノートでは、FPGA をアクティブ・パラレル (AP) コンフィギュレーション手法でコンフィギュレーションする前の、平行・フラッシュ・デバイスのプログラミングにおける、FPGA ベースの平行・フラッシュ・ローダ (PFL) の使用について説明します。

AP コンフィギュレーション手法では、業界標準平行・フラッシュ・デバイスを使用してアルテラ FPGA をコンフィギュレーションします。アルテラ Cyclone® III デバイスなどの高集積 FPGA の場合、平行・フラッシュ・デバイスを使用すると、平行・インタフェースによってコンフィギュレーション時間が短縮することに加え、コンフィギュレーション・データの格納に使用できるメモリ容量が増加します。しかし、平行・フラッシュ・デバイスは JTAG (Joint Test Action Group) インタフェースをサポートしていないため、JTAG による直接のデバイス・プログラミングもサポートしていません。

FPGA ベースの PFL を使用すれば、FPGA の JTAG インタフェースを介して平行・フラッシュ・デバイスのインシステム・プログラミングを行うことができます。AP コンフィギュレーション手法でアルテラ FPGA をコンフィギュレーションする前に、PFL によりフラッシュ・デバイスを間接的にプログラムすることが可能になります。



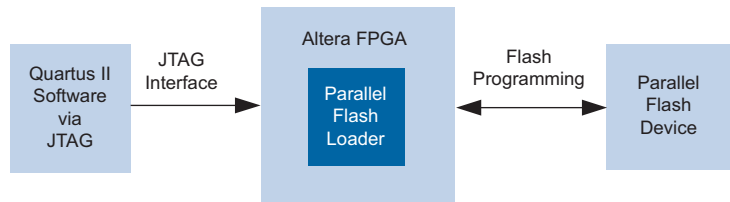
Cyclone III デバイスおよびサポートされる平行・フラッシュ・デバイスについて詳しくは、「Cyclone III デバイス・ハンドブック Volume 1」の「Cyclone III デバイスのコンフィギュレーション」の章を参照してください。

FPGA ベース の平行・ フラッシュ・ ローダ

FPGA ベースの PFL は、JTAG と平行・フラッシュ・インタフェースをブリッジする FPGA 内部のソフト IP (Intellectual Property) コアです。PFL により、JTAG インタフェースからのシリアル・プログラミング・ビットストリームを使用して、フラッシュ・データ、アドレス、およびフラッシュ・プログラミング用コントロール・ピンをコントロールすることができます。JTAG インタフェースは必要なピン数が少なく、フラッシュ・デバイスと同じインタフェースを共有するため、フラッシュ・プログラミング・プロセスが簡素化されます。

図 1 に、フラッシュ・プログラミングにおける PFL インタフェースと JTAG インタフェースおよび平行・フラッシュ・デバイスの関係を示します。

図1. フラッシュ・プログラミングにおける PFL インタフェース



PFL のプログラミング・フロー

FPGA に PFL をコンフィギュレーションし、フラッシュ・デバイスをプログラムするには、以下に示すステップの最初の 2 つを実行します。3 番目のステップはオプションです。

1. PFL のコンフィギュレーション

PFL によって FPGA をコンフィギュレーションして、JTAG と平行・フラッシュ・デバイス・インタフェースの間にブリッジを設定します。FPGA がすでに PFL によってコンフィギュレーションされている場合は、このステップを省略してもかまいません。

2. フラッシュ・プログラミング

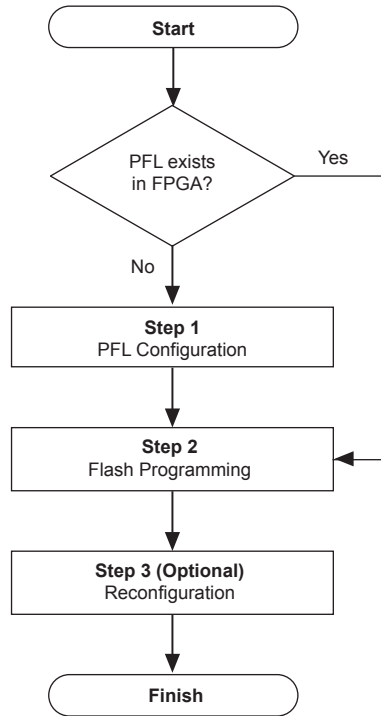
Quartus® II ソフトウェアからのシリアル・プログラミング・ビットストリームを使用して、JTAG インタフェースを介して平行・フラッシュ・デバイスをプログラムします。

3. リコンフィギュレーション (オプション)

フラッシュをプログラムした後、FPGA をリセットし、アップデートされたデザインを AP コンフィギュレーション手法によってコンフィギュレーションすることもできます。

図 2 に、PFL を使用したフラッシュ・デバイスのプログラミング・フローを示します。

図 2. PFL を使用したフラッシュ・デバイスのプログラミング



Quartus II ソフトウェアによるサポート

Quartus II ソフトウェアには、FPGA のコンフィギュレーション、パラレル・フラッシュ・デバイスのプログラミング、あるいはその両方を実行するための Quartus II Programmer ツールが用意されています。シングル・ステップ実行の場合、Programmer はまず FPGA をコンフィギュレーションしてからフラッシュ・デバイスのプログラミングを行います。

フラッシュ・デバイスをプログラムするには、以下のいずれかの方法を使用します。

- 「デフォルトの PFL」
- 「ユーザー・デザインの PFL」

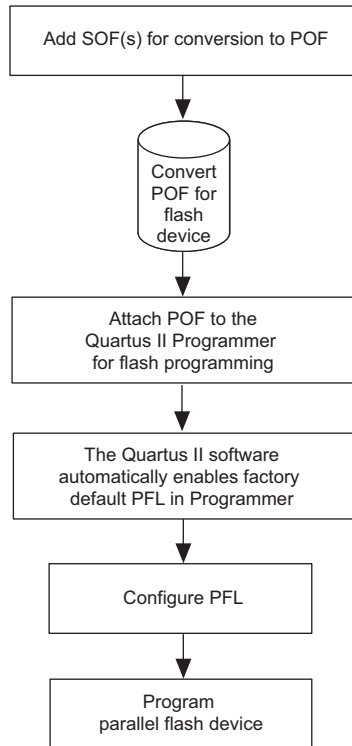
デフォルトの PFL

Quartus II ソフトウェアには、フラッシュ・プログラミング用のデフォルトの PFL が用意されています。

デフォルトの PFL を使用すると、PFL デザインの作成は不要になりますが、まず FPGA に PFL をコンフィギュレーションしないと、フラッシュ・デバイスのプログラミングは行えません。

図 3 に、デフォルトの PFL を使用する場合の Quartus II ソフトウェアにおけるプログラミング・フローを示します。

図 3. デフォルトの PFL を使用する場合のプログラミング・フロー

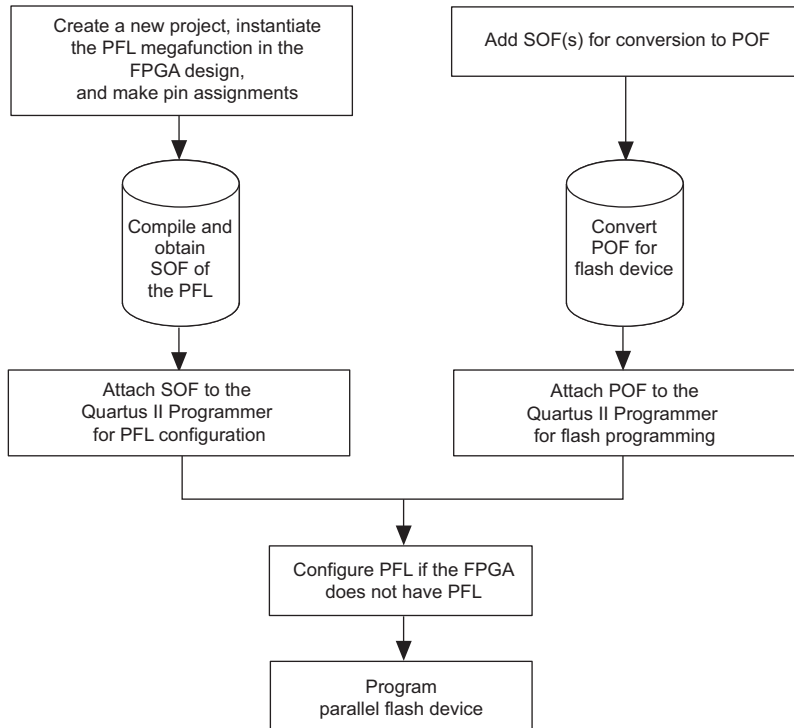


ユーザー・デザインの PFL

あるいは、ユーザー・デザインの PFL ロジックをインスタンス化することもできます。PFL プログラミング・ロジックは、Quartus II ソフトウェアで PFL メガファンクションを使用して生成することができます。ユーザー・デザインの PFL ロジックがすでに FPGA に格納されている場合、FPGA のリコンフィギュレーションは不要です。これにより、FPGA の動作を中断することなく、PFL でフラッシュをプログラムすることが可能になります。

図 4 に、ユーザー・デザインの PFL を使用する場合の Quartus II ソフトウェアにおけるプログラミング・フローを示します。


図 4. ユーザー・デザインの PFL を使用する場合のプログラミング・フロー



Quartus II ソフトウェア における PFL の使用

以下、Quartus II ソフトウェアにおける PFL の使用方法について、大きく 3 つのステップに分けて説明します。その後の項で、各ステップについてさらに詳しく説明します。

1. PFL メガファンクションによるユーザー・デザインの PFL ロジックのインスタンス化。この手順では、PFL の入力および出力ポートの機能についても説明します。
2. FPGA コンフィギュレーション・データを格納した SRAM オブジェクト・ファイル (SOF) から、パラレル・フラッシュ・プログラミング用プログラマ・オブジェクト・ファイル (POF) への変換
3. Quartus II Programmer で POF をパラレル・フラッシュ・デバイスにプログラミング

 このアプリケーション・ノートは、Quartus II ソフトウェア v7.1 以降を対象としていますが、Quartus II ソフトウェア v7.2 で取り込んだスクリーンショットを使って説明します。

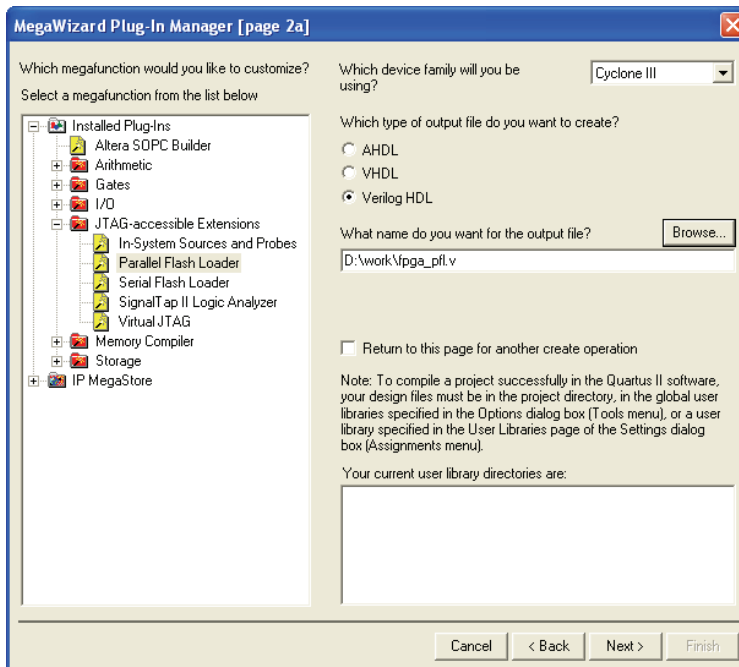
PFL メガファンクションによるユーザー・デザインの PFL ロジックのインスタンス化

以下の手順では、PFL メガファンクションによってユーザー・デザインの PFL ロジックをインスタンス化する方法について説明します。デフォルトの PFL を使用してフラッシュ・デバイスをプログラムする場合は、この手順を省略してもかまいません。

PFL のインスタンスを生成するには、以下のステップを実行します。

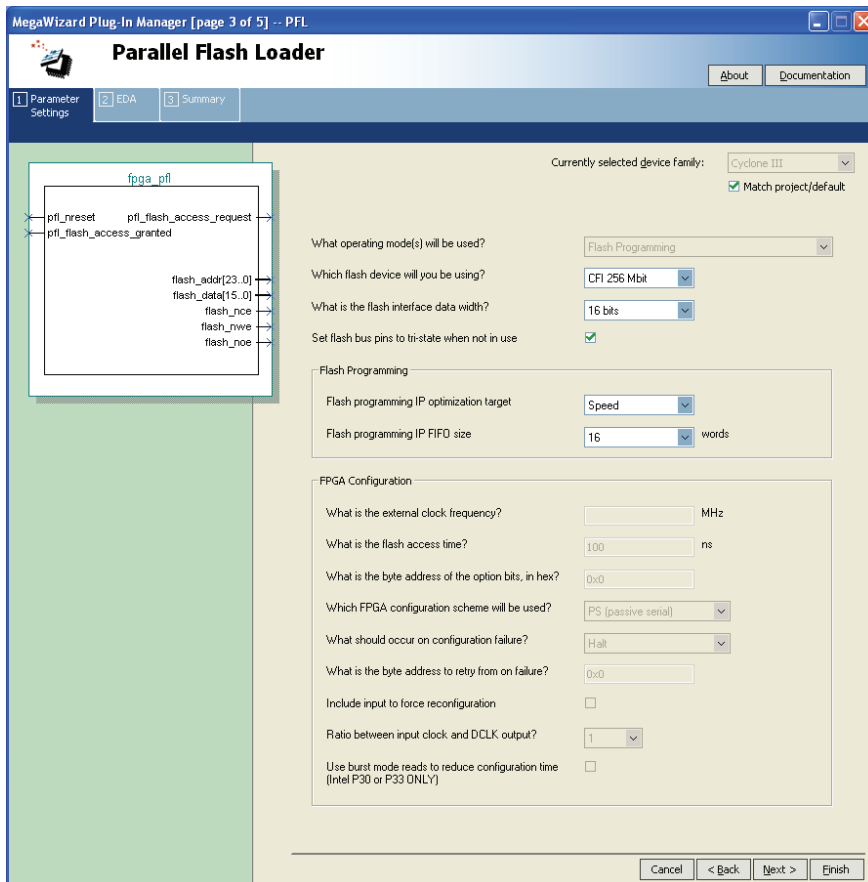
1. Quartus II ソフトウェアの Tools メニューの **MegaWizard Plug-In Manager** をクリックします。
2. page 1 の **Create a new custom megafunction variation** をクリックします。
3. **Next** をクリックします。page 2a が表示されます (図 5)。

図 5. MegaWizard Plug-In Manager [page 2a] ダイアログ・ボックス



4. メガファンクション・リストの **JTAG-accessible Extensions** フォルダを展開し、**Parallel Flash Loader** を選択します。
5. デバイス・ファミリ・リストから **Cyclone III** を選択します。
6. 出力ファイル・リストから HDL 出力ファイル・タイプを選択します (図 5 の例では Verilog HDL が選択されています)。
7. <プロジェクト・ディレクトリ>\<ファイル名> の形式で出力ファイル名を指定します。
8. **Next** をクリックします。page 3 が表示されます (図 6)。

図 6. PFL メガファンクションのパラメータ設定

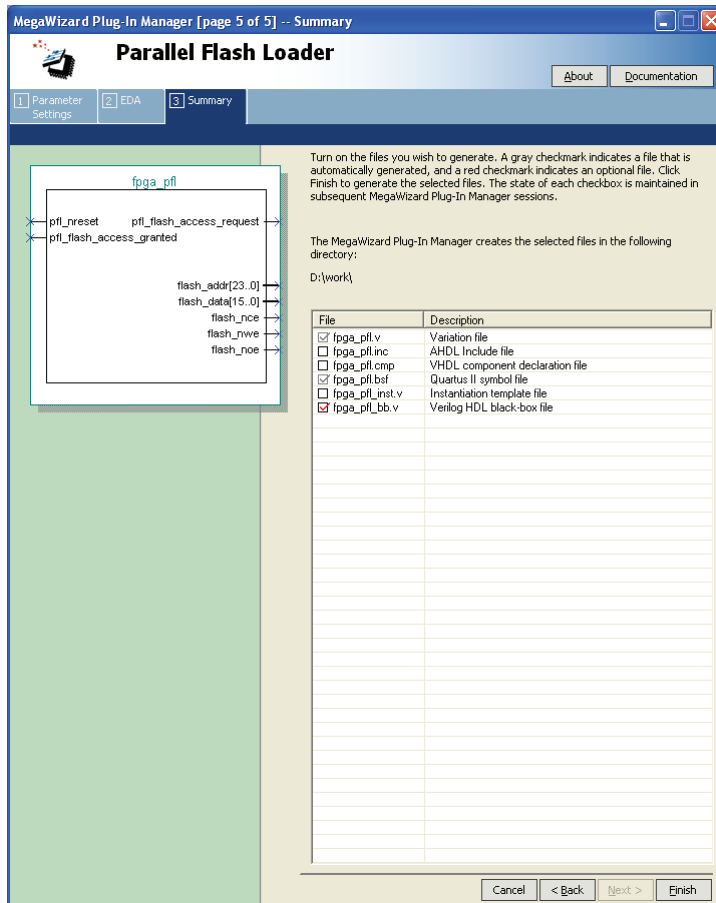


9. 表 1 に示す PFL メガファンクションのパラメータ値を指定します。FPGA ベースの PFL の場合、Operating Mode パラメータ設定はグレー表示されます。

メガファンクションのパラメータ	説明
Flash device	プログラムするデバイスの容量。
Flash interface data width	プログラムするデバイスのデータ幅。
Tri-state flash bus	PFL がフラッシュ・デバイスにアクセスする必要がないときに、フラッシュ・デバイスとインタフェースするすべてのピンをトライステートにします。
Flash programming IP optimization target	フラッシュ・プログラミング IP は、速度または論理規模の最適化が可能です。速度が最適化された IP では、フラッシュ・プログラミングの所要時間は短くなりますが、メガファンクションが使用するロジック・エレメントが増加します。論理規模が最適化された IP では、IP に必要なロジック・エレメントは少なくなります。フラッシュ・プログラミングの所要時間が長くなります。
FIFO size	速度が最適化されたフラッシュ・プログラミング IP の場合、PFL は追加のロジック・エレメントを使用して、フラッシュ・プログラミング中に使用されるプログラミング・データの一時的なストレージとして FIFO を実装します。FIFO サイズのオプションは 16 ワードまたは 32 ワードです。FIFO サイズを大きくするとプログラミング時間が短縮されます。

10. **Next** をクリックします。page 4 が表示され、PFL メガファンクションに必要なシミュレーション・ファイルがリストされます。PFL にはシミュレーション・ファイルがなく、シミュレートが行えないため、シミュレーション・ファイルはリストされません。
11. **Next** をクリックします。page 5 の Summary ページが表示されます (図 7)。このページには、メガファンクションに対して生成されるファイルが表示されます。追加で生成したいファイル・タイプがあれば選択します。

図7. PFL メガファンクションのサマリ



12. **Finish** をクリックします。Quartus II ソフトウェアにより、page 2a で選択した HDL ファイル形式の PFL メガファンクションと、page 5 で選択した追加のファイルが生成されます。

PFL メガファンクションの入力および出力ポート

この項では、PFL メガファンクションの入力および出力ポートの機能について説明します。図 8 に、PFL メガファンクションのシンボルを示します。

図 8. PFL メガファンクションのシンボル

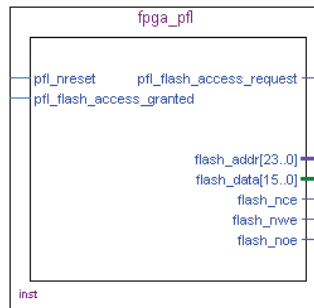


表 2 に、PFL の入力および出力ポートの機能を示します。

表 2. PFL の入力および出力ポートの機能		
ポート	タイプ	機能
pfl_nreset	入力	リセット・ピン。Low にプルダウンすることによって PFL をリセットします。
pfl_flash_access_granted	入力	システム・レベルの同期化に使用されます。このピンは、フラッシュ・デバイスへのアクセスをコントロールする外部ホストによってドライブすることができます。PFL をフラッシュ・マスタとして使用したい場合は、このアクティブ High ピンを永続的に High にプルアップします。Low にプルダウンすることにより、JTAG からフラッシュ・デバイスへのアクセスを防止します。
pfl_flash_access_request	出力	システム・レベルの同期化に使用されます。このピンは、必要に応じて外部ホストに接続することができます。PFL は、JTAG がフラッシュ・デバイスにアクセスしているときは常にこのピンを High にドライブします。
flash_addr[23..0]	出力	FPGA の PADD[23..0] バスに接続します。
flash_data[15..0]	出力	FPGA の DATA[15..0] バスに接続します。
flash_nce	出力	FPGA の flash_nCE ピンに接続します。Low 信号によってフラッシュ・デバイスをイネーブルします。
flash_nwe	出力	FPGA の nWE ピンに接続します。書き込み動作中に nWE ピンを Low にドライブすることにより、DATA[15..0] バス上のデータが有効であることをフラッシュ・デバイスに示します。
flash_noe	出力	FPGA の nOE ピンに接続します。読み出し動作中に nOE ピンを Low にドライブすることにより、DATA[15..0] バス上のフラッシュ・デバイス出力をイネーブルします。

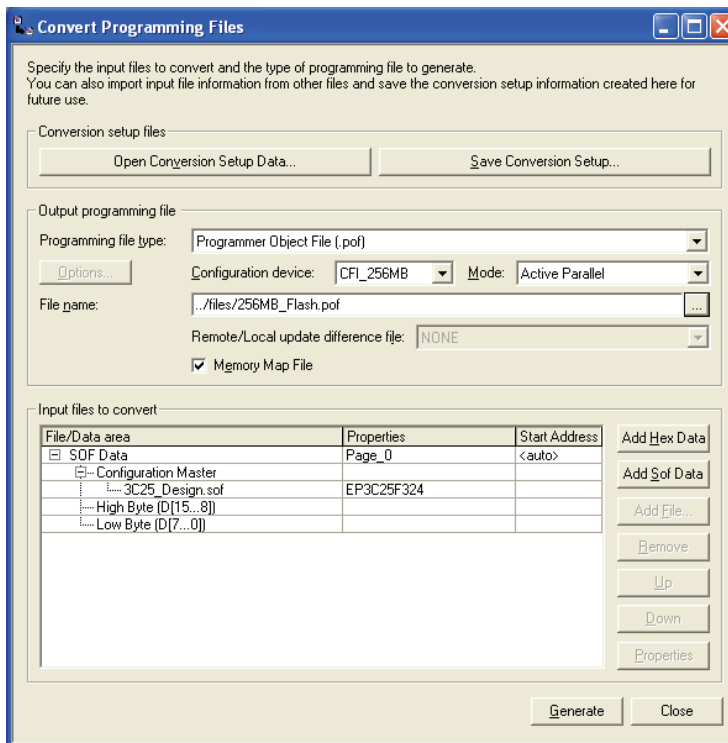
SRAM オブジェクト・ファイルからパラレル・フラッシュ・デバイス用プログラマ・オブジェクト・ファイルへの変換

フラッシュ・デバイス用 POF を作成するには、FPGA デバイスから生成された SOF を使用します。フラッシュ・デバイス用 POF を作成するときに、ユーザー・データを格納した 16 進 (HEX) ファイルを選択することにより、コンフィギュレーション以外のデータを POF に追加することもできます。

複数の SOF を 1 つの POF にまとめるには、以下のステップを実行します。

1. Quartus II ソフトウェアの File メニューの **Convert Programming Files** をクリックします。 **Convert Programming Files** ダイアログ・ボックスが表示されます (図 9)。

図 9. シングル・デバイス・コンフィギュレーション・チェーン用プログラミング・ファイルの変換



2. **Programming file type** リストから **Programmer Object File (.pof)** を選択します。

3. **Configuration device** リストから、正しい容量のコモン・フラッシュ・インタフェース (CFI) デバイスを選択します。例えば、CFI_256 は容量が 256M ビットの平行・フラッシュ・メモリを表します。
4. コンフィギュレーション手法として、**Mode** リストから **Active Parallel** を選択します。
5. **File name** ボックスで出力ファイル名を指定します。


Input files to convert の下を見ると、**SOF Data** 階層があり、**Configuration Master**、**Low Byte [D[7...0]]**、および **High Byte [D[15...8]]** まで展開されています。

6. シングル・デバイス・コンフィギュレーション・チェーン用 SOF を追加するには、**Configuration Master** を選択し、**Add File** をクリックします。
7. 追加する SOF を選択し、**Open** をクリックします。

マルチ・デバイス・コンフィギュレーション・チェーンの場合は、同じページに複数の SOF を追加することができます。SOF の順序は、チェーン内でのデバイスの順序に合わせる必要があります。

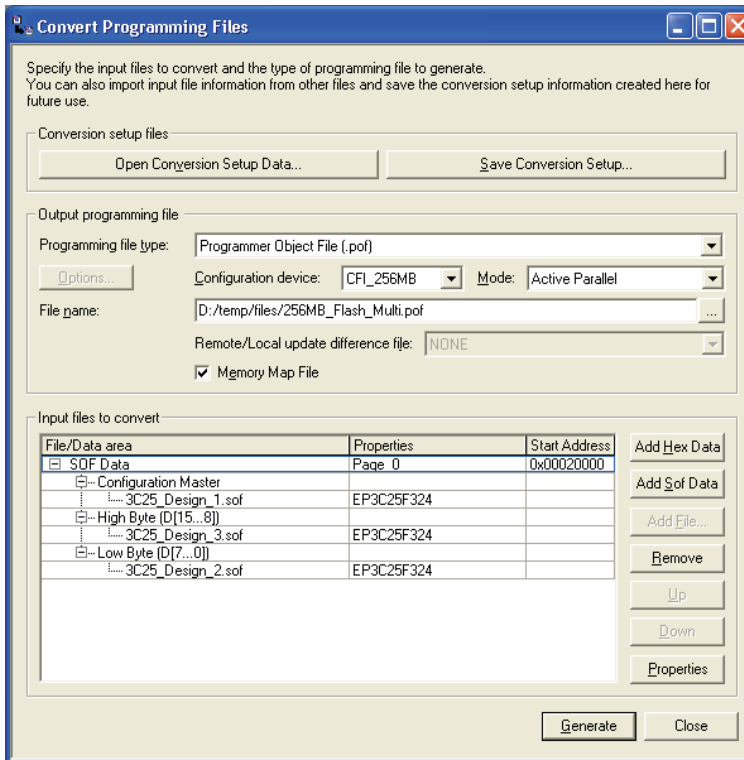
バイト幅のマルチ・デバイス・コンフィギュレーション・チェーン内のデバイスの場合、チェーン内でのデバイスの順序に従って、**Low Byte [D[7...0]]** に SOF を追加します。

ワード幅のマルチ・デバイス・コンフィギュレーション・チェーン内のデバイスの場合は、[図 10](#) に示すように、**DATA [7..0]** および **DATA [15..8]** バスにそれぞれ接続されたデバイスの順序に従って、**Low Byte [D[7...0]]** および **High Byte [D[15...8]]** に SOF を追加します。

 ワード幅のデバイス・コンフィギュレーションが使用できるのは、Quartus II ソフトウェア v7.2 以降でのみです。

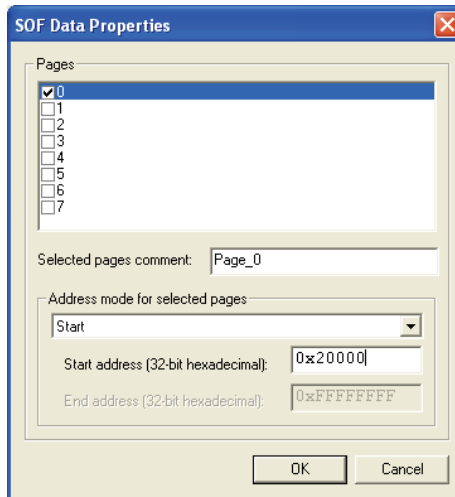
別のページに SOF データを格納したい場合は、**Add SOF Data** をクリックします。**Input files to convert** リストの下に、新しい **SOF_Data** 行が表示されます。そこで新しいページの SOF を追加します。

図 10. ワード幅のマルチ・デバイス・コンフィギュレーション・チェーン用プログラミング・ファイルの変換



8. SOF_Data のページ番号およびページ名を設定するには、SOF_Data を選択し、**Properties** をクリックします。**SOF Data Properties** ダイアログ・ボックスが表示されます (図 11)。

図 11. SOF Data Properties ダイアログ・ボックス



9. **Address mode for selected pages** リストから **Start** を選択します。
10. **Start address** ボックスに 0x20000 と入力することにより、開始アドレスをバイト・アドレッシングで指定します。



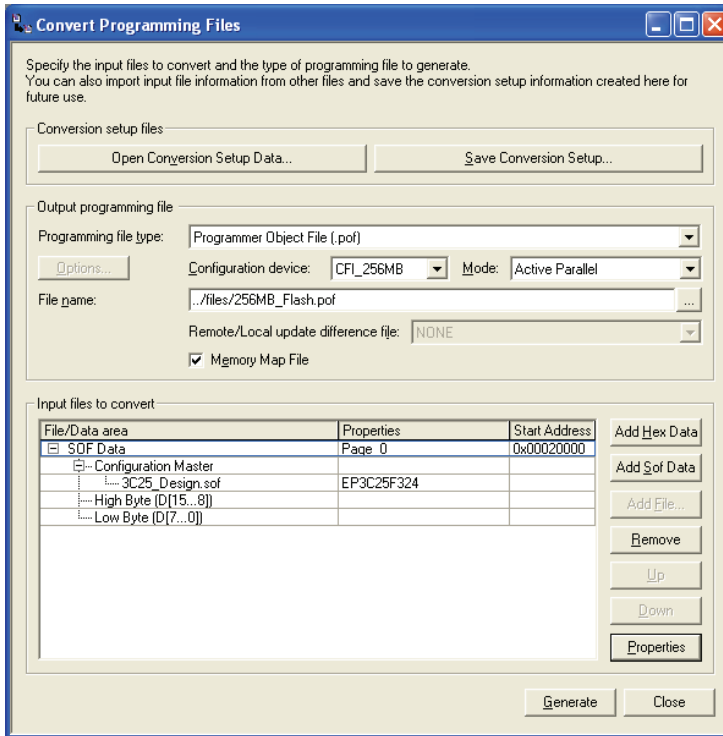
Cyclone III の AP コンフィギュレーション手法では、デフォルトのブート・アドレス（バイト・アドレッシングを使用する場合は 0x20000、またはワード・アドレッシングを使用する場合は 0x10000）からコンフィギュレーションします。別のブート・アドレスから AP コンフィギュレーション手法を構成するには、JTAG インストラクション APFC_BOOT_ADDR を実行して、Cyclone III デバイスのブート・アドレスを変更します。



JTAG インストラクションの使用方法について詳しくは、「Cyclone III デバイス・ハンドブック Volume 1」の「Cyclone III デバイスのコンフィギュレーション」の章を参照してください。

11. **OK** をクリックします。図 12 に示すように、SOF データの Start Address に 0x20000 と表示されます。

図 12. バイト・アドレッシングを使用した場合の開始アドレス (0x20000)



12. あるいは、HEX ファイル内のユーザー・データを格納することもできます。HEX ファイル内のユーザー・データを格納するには、以下のステップを実行します。
 - a. **Input files to convert** の **Add Hex Data** をクリックします。**Add Hex Data** ダイアログ・ボックスが表示されます。
 - b. **Addressing mode** で、必要なアドレッシング・モードを選択します。**Set start address** をオンにして、開始アドレスを指定します。
 - c. **Hex file** ボックスで HEX ファイル名を指定します。
 - d. **OK** をクリックします。



HEX ファイルのみを使用して、フラッシュ・デバイス用 POF を作成することはできません。POF を作成するときには、FPGA 用 SOF も追加しなければなりません。

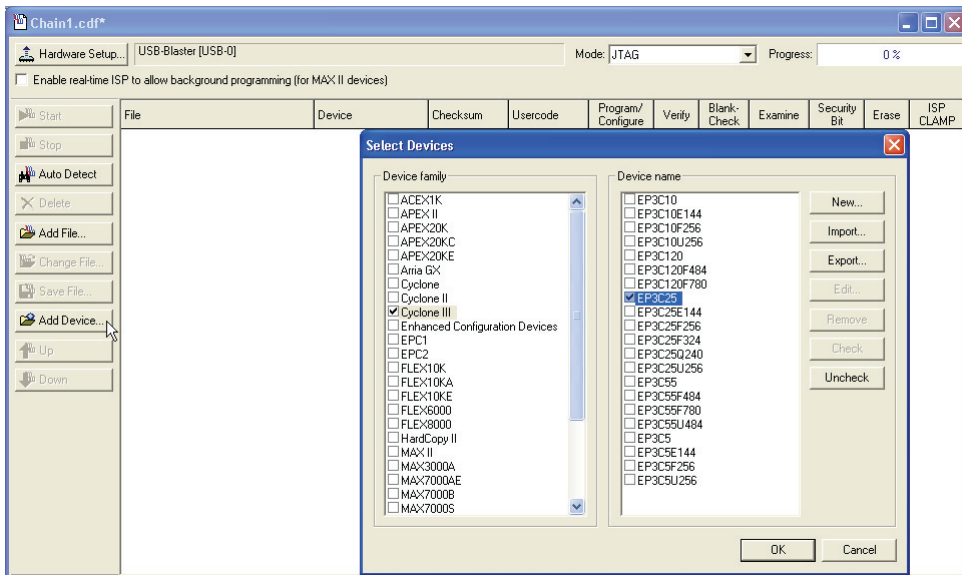
13. **Generate** をクリックすると、POF が作成されます。

デフォルトの PFL を使用した平行・フラッシュ・デバイスのプログラミング

Quartus II Programmer でフラッシュ・デバイスをプログラムするには、以下のステップを実行します。

1. Quartus II ソフトウェアの Tools メニューの **Programmer** をクリックします。
2. Programmer ウィンドウの **Add Device** をクリックします。図 13 に示すように、**Select Devices** ダイアログ・ボックスが表示されます。

図 13. PFL 用 SOF の追加



3. **Device Name** で、追加するデバイス名を選択します。
4. **OK** をクリックします。そのデバイス名が Programmer ウィンドウに表示されます。
5. 図 14 に示すように、追加したデバイス名を選択して右クリックし、**Attach Flash Device** をクリックします。Select Flash Device ダイアログ・ボックスが表示されます (図 15)。

図 14. フラッシュ・デバイスの付加

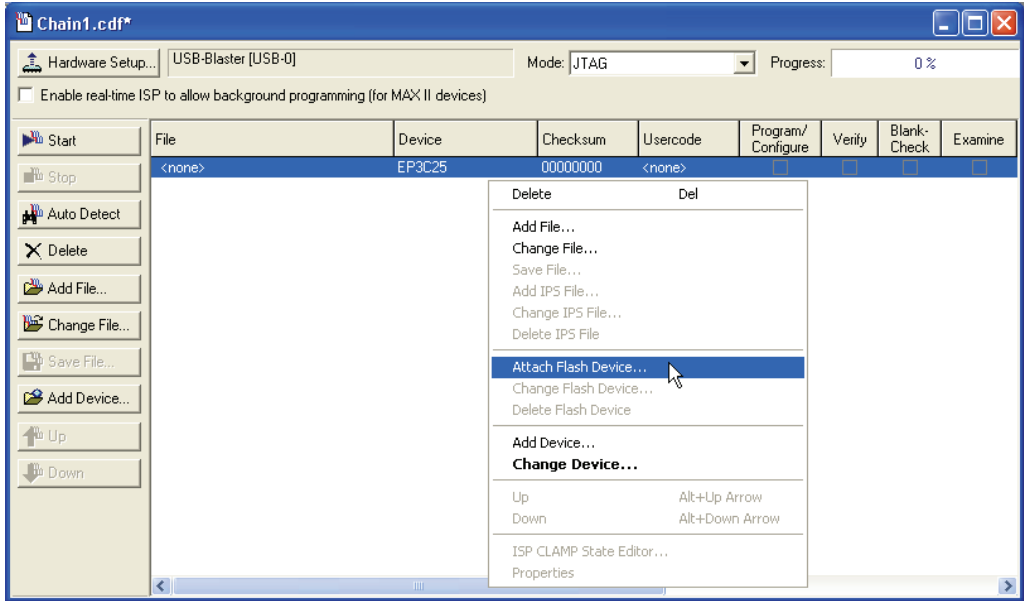
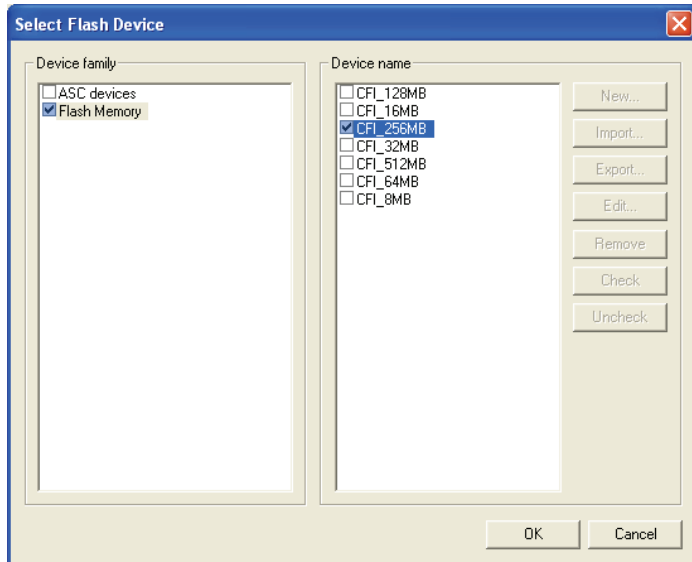
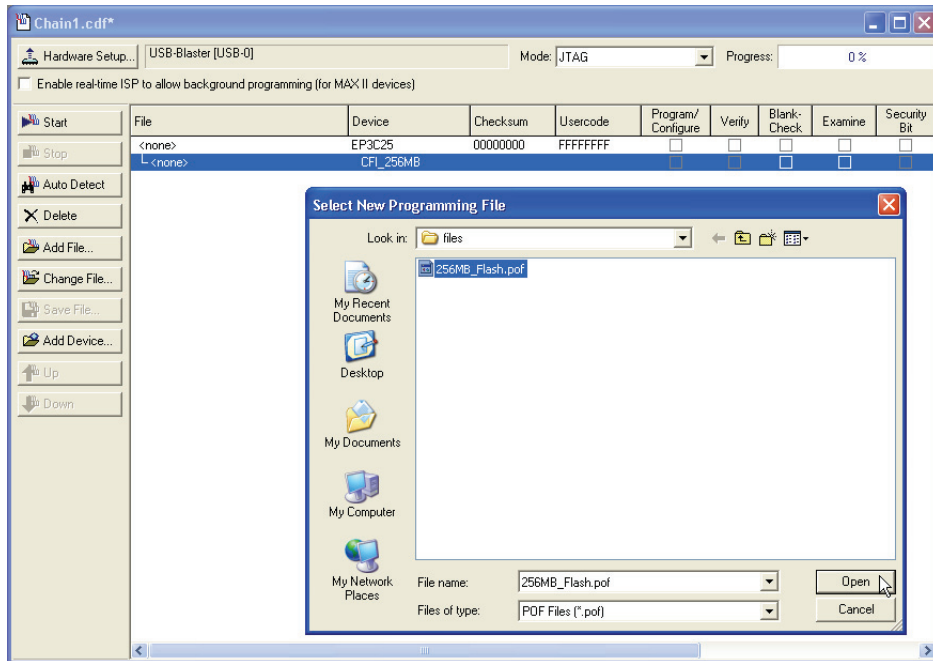


図 15. フラッシュ・デバイスの選択



6. **Device family** の **Flash Memory** をオンにします。
7. **Device name** でフラッシュ・デバイスの容量を選択します。
8. **OK** をクリックして Programmer ウィンドウに戻ります。
9. そのデバイス名を選択し、右クリックします。**Change File** をクリックします。**Select New Programming File** ダイアログ・ボックスが表示されます (図 16)。

図 16. フラッシュ・プログラミング・デバイス用 POF の追加



10. フラッシュ・デバイス用 POF を選択し、**Open** をクリックします。



Quartus II Programmer では、一度にチェーン内の 1 つの FPGA にしかフラッシュ・デバイス用 POF を付加できないため、一度にプログラムできるのはチェーン内の 1 つのフラッシュ・デバイスのみです。チェーンに含まれる別の FPGA のフラッシュ・デバイスをプログラムするには、最初の FPGA のフラッシュ・デバイス用 POF を削除した上で、チェーンに含まれる次の FPGA のフラッシュ・デバイス用 POF を追加しなければなりません。

11. **Program/Configure** カラムで、追加した POF の **Page_0** に対応するチェック・ボックスをオンにします。

図 17 に示すように、Quartus II Programmer によってデフォルトの PFL イメージが自動的に有効になります。PFL コンフィギュレーションをバイパスするには、図 18 に示すように、

Program/Configure カラムの下の対応するチェック・ボックスをオフにすることにより、デフォルトの PFL イメージを無効にします。


 フラッシュ・デバイス全体を消去またはプログラムするには、その POF に対応するチェック・ボックスをオンにします。フラッシュ・デバイスの特定のページを消去またはプログラムするには、そのページに対応するチェック・ボックスをオンにします。

図 17. デフォルトの PFL イメージとフラッシュ・デバイス用 POF を表示した状態の Quartus II Programmer

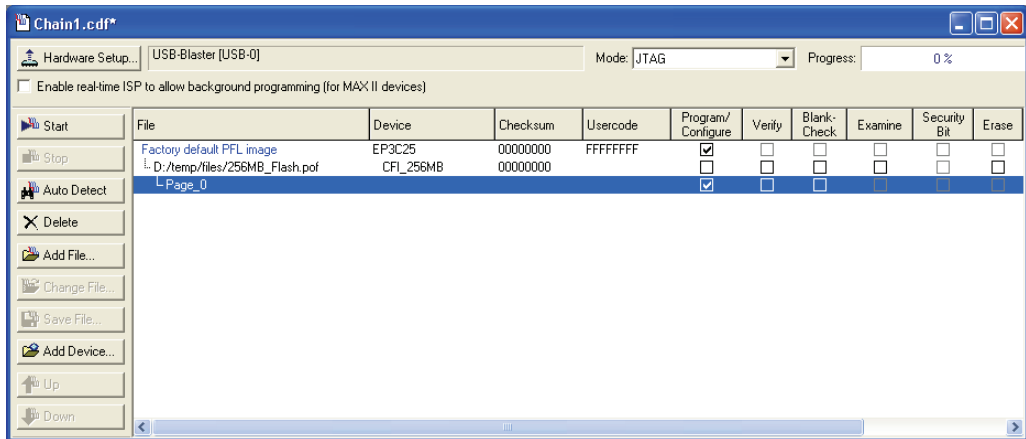
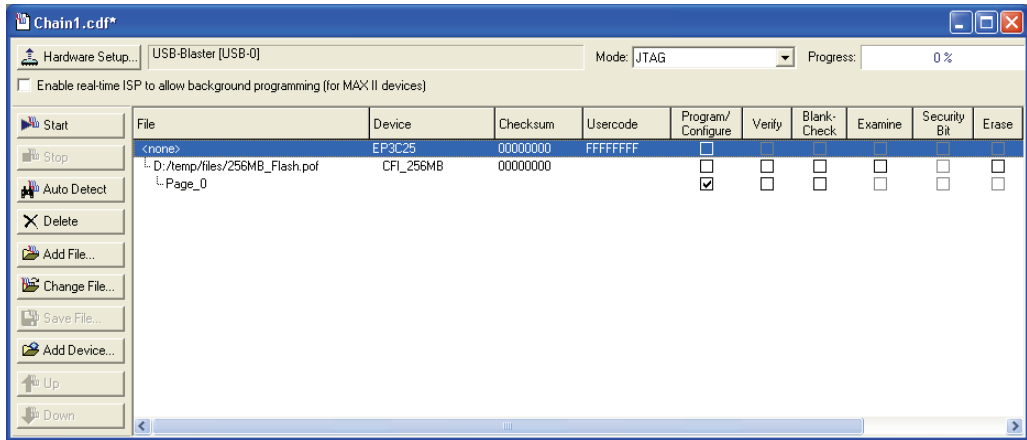


図 18. デフォルトの PFL イメージの無効化



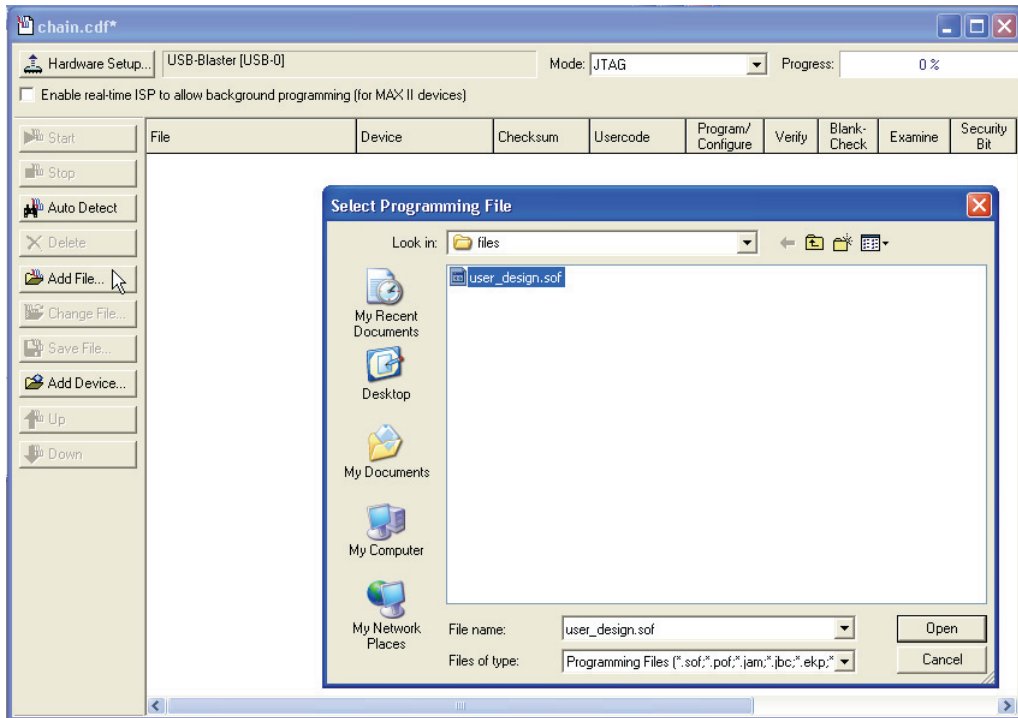
12. **Start** をクリックすると、PFL のコンフィギュレーションおよびフラッシュ・デバイスのプログラミングを行うことができます。

ユーザー・デザインの PFL を使用した平行・フラッシュ・デバイスのプログラミング

Quartus II Programmer でフラッシュ・デバイスをプログラムするには、以下のステップを実行します。

1. Quartus II ソフトウェアの Tools メニューの **Programmer** をクリックします。
2. Programmer ウィンドウの **Add File** をクリックします。図 19 に示すように、**Select Programming File** ダイアログ・ボックスが表示されます。

図 19. PFL 用 SOF の追加



3. PFL ロジックを格納したユーザー・デザインの SOF を選択します。
4. **Open** をクリックします。その SOF 名が Programmer ウィンドウに表示されます。
5. 追加した SOF を選択し、右クリックします。図 20 に示すように、**Attach Flash Device** をクリックします。**Select Flash Device** ダイアログ・ボックスが表示されます (図 21)。

図 20. フラッシュ・デバイスの付加

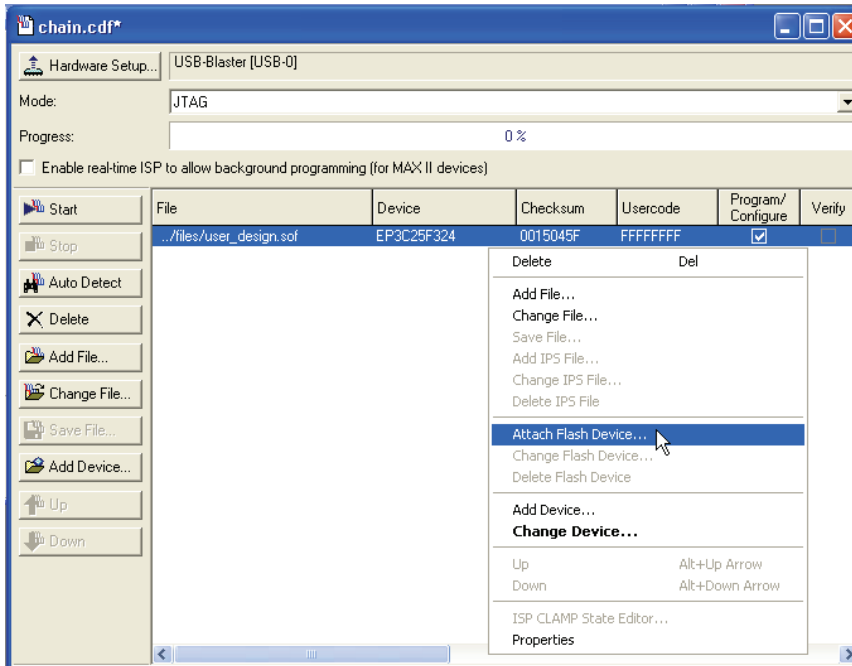
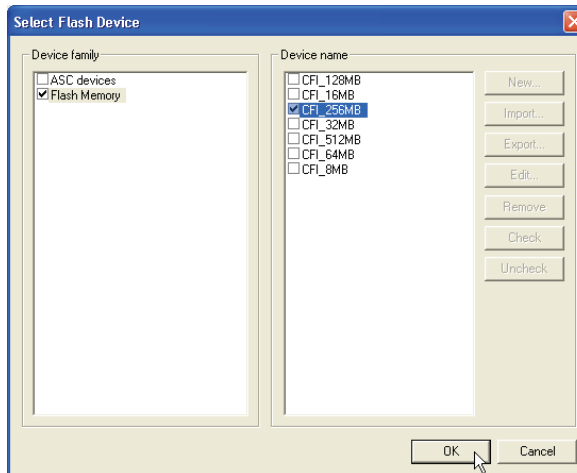
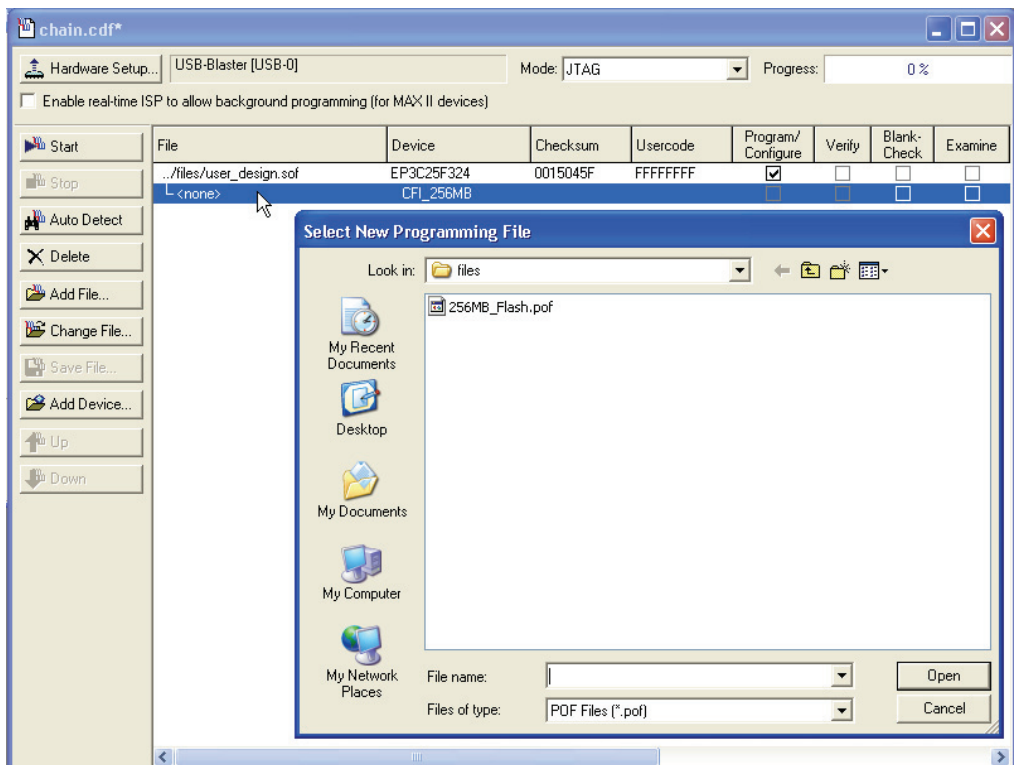


図 21. フラッシュ・デバイスの選択



6. **Device family** の **Flash Memory** をオンにします。
7. **Device name** でフラッシュ・デバイスの容量を選択します。
8. **OK** をクリックして Programmer ウィンドウに戻ります。
9. そのデバイス名を選択し、右クリックします。 **Change File** をクリックします。 **Select New Programming File** ダイアログ・ボックスが表示されます (図 22)。

図 22. フラッシュ・プログラミング・デバイス用 POF の追加



10. フラッシュ・デバイス用 POF を選択し、 **Open** をクリックします。



Quartus II Programmer では、一度にチェーン内の 1 つの FPGA にしかフラッシュ・デバイス用 POF を付加できないため、一度にプログラムできるのはチェーン内の 1 つのフラッシュ・デバイスのみです。チェーンに含まれる別の FPGA のフラッシュ・デバイスをプログラムするには、最初の FPGA のフラッシュ・デバイス用 POF を削除した上で、チェーンに含まれる次の FPGA のフラッシュ・デバイス用 POF を追加しなければなりません。

11. 図 23 に示すように、**Program/Configure** カラムで、SOF および追加した POF の **Page_0** に対応するチェック・ボックスをオンにします。この設定により、Quartus II Programmer はフラッシュ・デバイスのプログラミングの前に、まず PFL ロジックを格納したユーザー・デザインの SOF をコンフィギュレーションします。PFL コンフィギュレーションをバイパスするには、図 24 に示すように、**Program/Configure** カラムの下の対応するチェック・ボックスをオフにすることにより、SOF を無効にします。



フラッシュ・デバイス全体を消去またはプログラムするには、その POF に対応するチェック・ボックスをオンにします。フラッシュ・デバイスの特定のページを消去またはプログラムするには、そのページに対応するチェック・ボックスをオンにします。

図 23. ユーザー・デザインの PFL イメージとフラッシュ・デバイス用 POF を表示した状態の Quartus II Programmer

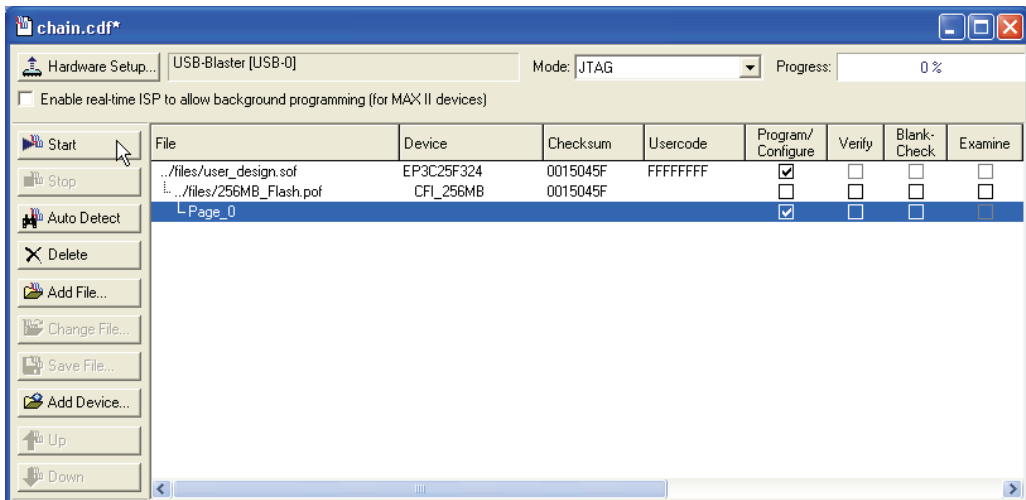
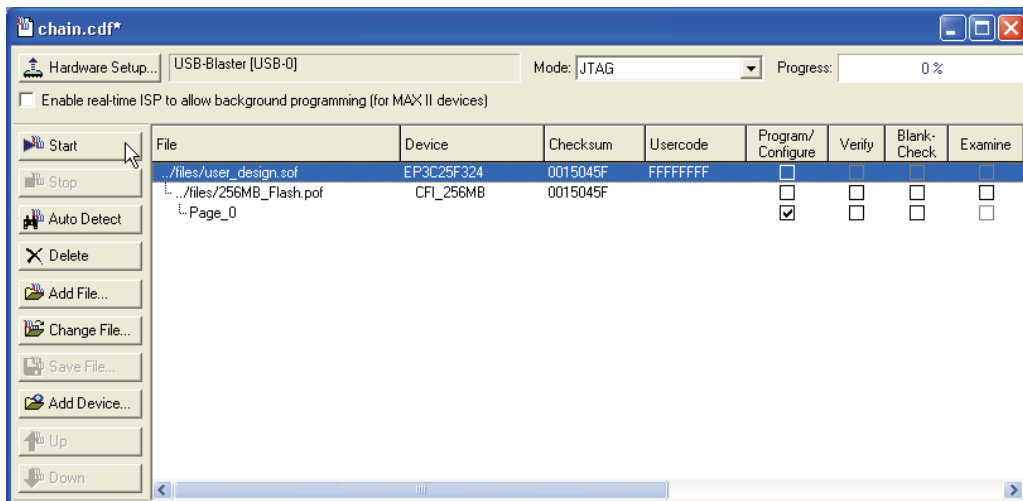


図 24. ユーザー・デザインの PFL イメージの無効化



12. **Start** をクリックすると、PFL のコンフィギュレーションおよびフラッシュ・デバイスのプログラミングを行うことができます。

Quartus II Programmer では、FPGA に PFL が内蔵されている場合、コンフィギュレーション・データ・ページとユーザー・データ・ページのプログラム、検証、消去、またはブランク・チェックを個別に行うことができます。PFL コンフィギュレーションがすでに FPGA に格納されている場合は、PFL コンフィギュレーション・ステップをバイパスすることができます。

まとめ

AP コンフィギュレーション手法をサポートしたアルテラ FPGA に搭載された PFL 機能を使用すれば、インシステム・プログラミングによるパラレル・フラッシュ・デバイスのプログラミングが可能です。Quartus II ソフトウェアには、FPGA の JTAG インタフェースを介してパラレル・フラッシュ・デバイスをプログラムするのに必要なツールが用意されています。

参考資料

このアプリケーション・ノートでは、以下のドキュメントを参照しています。

- 「Cyclone III デバイス・ハンドブック Volume 1」の「Cyclone III デバイスのコンフィギュレーション」の章

改訂履歴

表 3 に、このアプリケーション・ノートの改訂履歴を示します。

日付およびドキュメント・バージョン	変更内容	概要
2007 年 12 月 v1.0	初版	—



101 Innovation Drive
San Jose, CA 95134
www.altera.com
Technical Support:
www.altera.com/support/
Literature Services:
literature@altera.com

Copyright © 2007 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.



I.S. EN ISO 9001