Quad SPI Flash メモリからの Nios II ブート方法 MAX 10 FPGA Development Kit 編

ver.15

2015 年 12 月 Rev.1

Quad SPI Flash メモリからの Nios II ブート方法

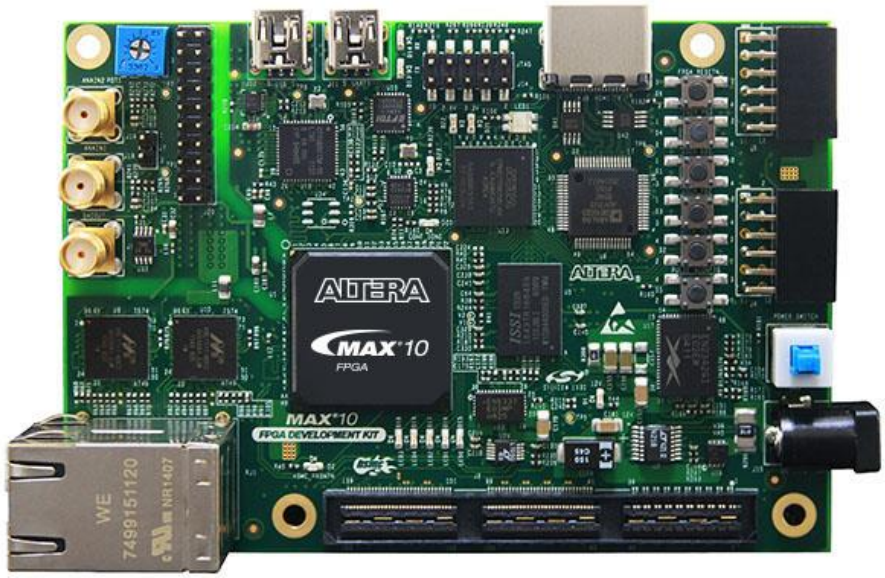
目次

1. はじめに

この資料は、MAX® 10 FPGA Development Kit に実装された 512Mb の Quad SPI Flash メモリに Nios® II のプログラムを格納し、MAX 10 デバイス内のデザインに実装した Altera Generic QUAD SPI Controller 経由で Nios II ブートを行う方法について記載した資料です。Quad SPI Flash メモリを外部不揮発性メモリとして使用して Nios II のプログラムを格納するためのサンプル・デザインとして参照ください。

1-1. 環境

評価用ボード ： アルテラ社 MAX 10 FPGA Development Kit（Rev.C 以降）



ハードウェア開発ツール ： アルテラ社 Quartus® Prime v15.1.0（以降、Quartus Prime と省略）

ソフトウェア開発ツール ： アルテラ社 Nios II Software Build Tools for Eclipse（以降、Nios II SBT と省略）

1-2. デザイン

1-2-1. スタンダード・デザイン

Nios II が実装された標準的なデザインです。Nios II Processor、Altera Generic QUAD SPI Controller、DDR3 SDRAM Controller、On-Chip Memory、Interval Timer、JTAG UART、PIO 等がインスタンスされています。本資料では、Nios II から PIO 経由で LED を点灯、JTAG UART 経由でキャラクタ出力を実行するためのデザインとして使用します。

1-2-2. パラレル・フラッシュ・ローダ・デザイン（PFL デザイン）

Nios II SBT で作成した Nios II のソフトウェア・プログラムを PFL 経由で Quad SPI Flash メモリに書き込むためのデザインです。

1-3. 処理フロー

① MAX 10 FPGA Development Kit 上でスタンダード・デザイン上の Nios II のプログラムが正常に動作することを JTAG ポート経由で確認します。

② Nios II SBT と Quartus Prime の Convert Programming Files を使用して Quad SPI Flash メモリに書き込むための POF ファイルを生成します。

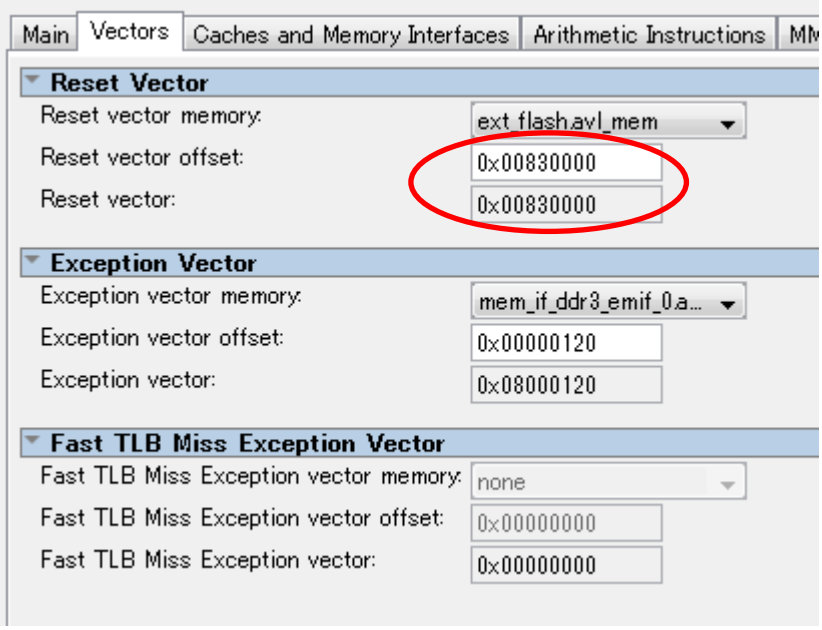
③ MAX 10 FPGA Development Kit の MAX 10 FPGA デバイスに PFL デザインを書き込み、Quartus Prime Programmer より ② で生成した POF ファイルを指定して、Quad SPI Flash メモリに Nios II ソフトウェア・プログラム・ファイルを書き込みます。

④ ① で生成したスタンダード・デザインの SOF ファイルを MAX 10 デバイスに書き込み、Quad SPI Flash メモリの該当アドレスから正しく Nios II ソフトウェア・プログラムが読み込まれ、ブートすることを確認します。

2. スタンダード・デザインの動作確認

2-1. Qsys の確認

スタンダード・デザインを Quartus Prime で開きます。次に、q_sys.qsys ファイルを Qsys で開きます。cpu をハイライトし、右クリック ⇒ Edit メニューで Nios II Processor を開きます。Vectors タブを見ると、以下のようになっています。

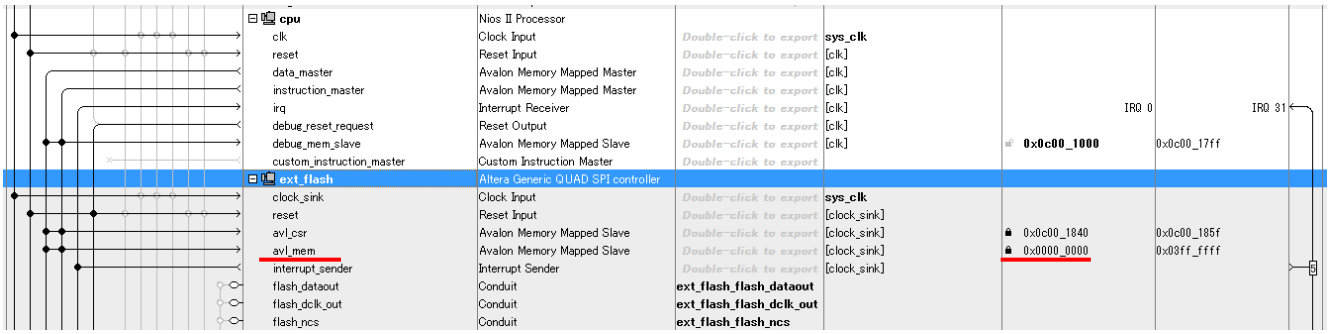


ここにある 0x00830000 アドレスが、Quad SPI Flash メモリ上のブート・アドレスになります。このアドレスの位置を先頭にして、Nios II のソフトウェア・プログラムを書き込む必要があります。この値は任意ですが、ここでは、MAX 10 FPGA Development Kit ユーザガイドの User Software の位置を指定しました。

Table 4-24: Default Memory Map of the 512-Mb QSPI Flash

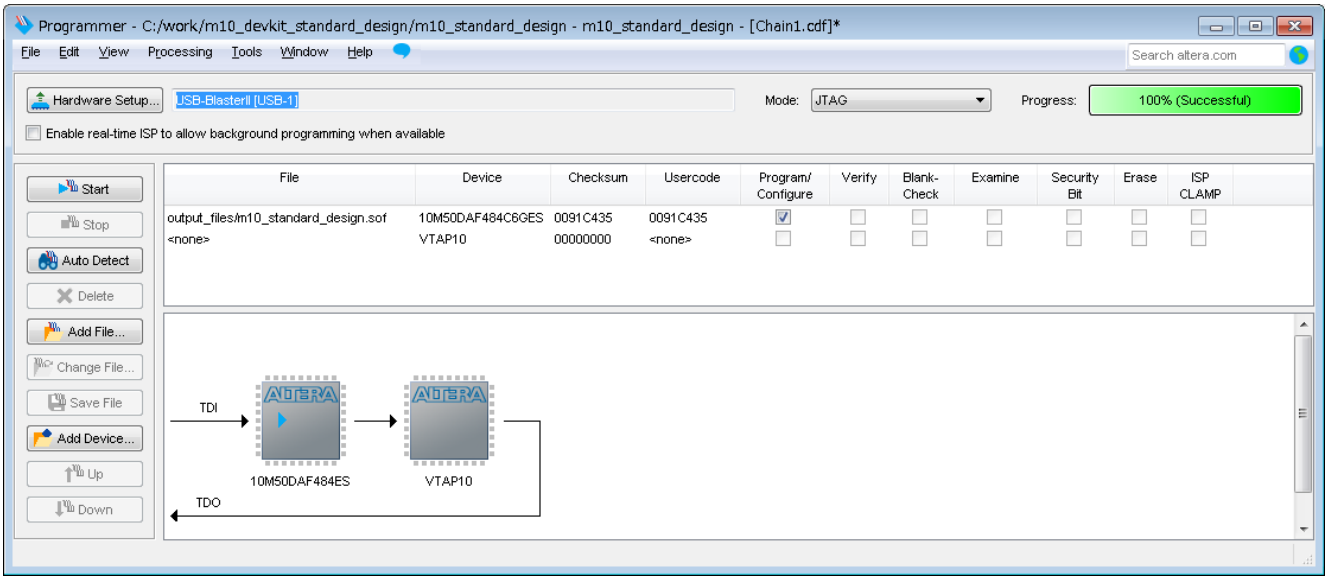
Block Description	Size (KB)	Address Range
Board test system scratch	512	0x03F8.0000 – 0x03FF.FFFF
User software	56640	0x0083.0000 – 0x03F7.FFFF
Factory software	4096	0x0043.0000 – 0x0082.FFFF
Zips(html, web content)	4096	0x0003.0000 – 0x0042.FFFF
Board information	64	0x0002.0000 – 0x0002.FFFF
Ethernet option bits	64	0x0001.0000 – 0x0001.FFFF
User design reset vector	64	0x0000.0000 – 0x0000.FFFF

Nios II Processor を閉じて、Qsys の System Contents タブに戻ります。ext_flash.avl_mem が Altera Generic QUAD SPI Controller のデータ・ポートとなります。Base アドレスが、0x0000_0000 となっているので、Nios II からは、ゼロ・オフセットでアクセスします。Quad SPI Flash メモリ上の 0x00830000 に書き込まれたソフトウェア・プログラムは、このポートを経由して Nios II に読み込まれます。

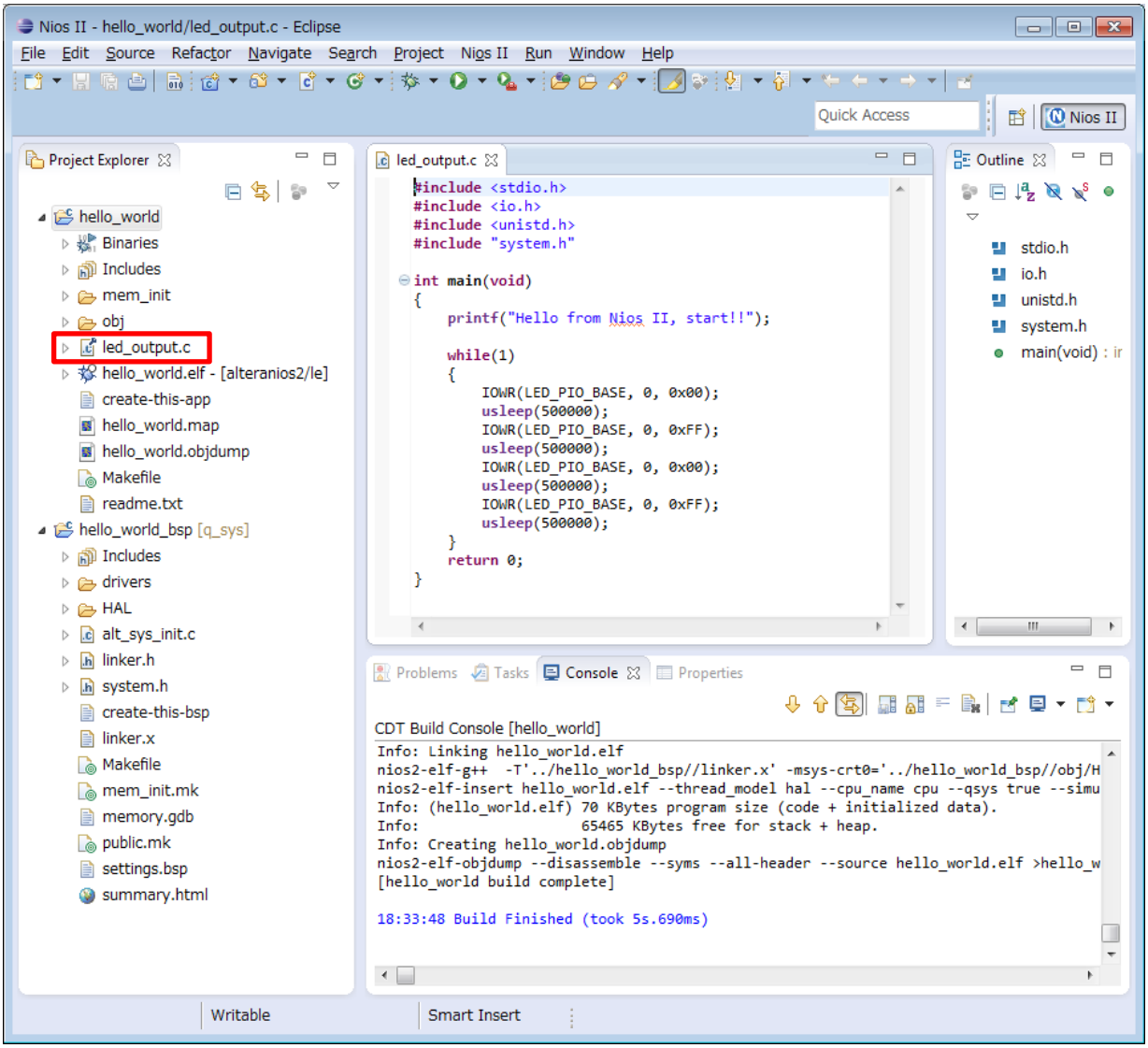


2-2. Nios II SBT での動作確認

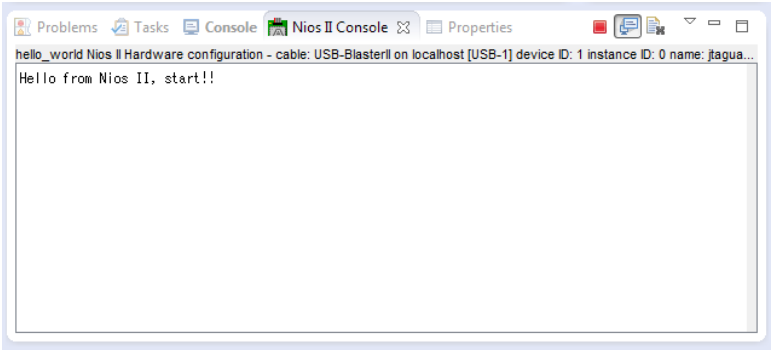
Qsys を閉じます。Programmer を起動し、SOF ファイルを書き込みます。



Nios II SBT を起動し、software フォルダ以下にある led_output.c ファイルを元にソフトウェア・プロジェクトを作成します。以下の図では、hello_world というプロジェクト名でプロジェクトを作成し、ビルドまで完了した状態です。



ビルドが正常に終わったら、ソフトウェア・プログラムを実行します。正常に実行されると、Nios II Console に以下のような表示が出力され、ボード上では、LED0,1,2,3 が同じタイミングで点滅します。

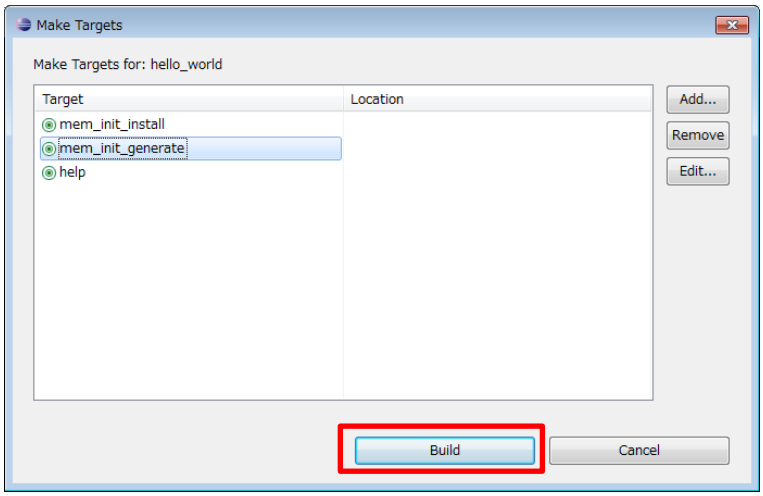


3. Quad SPI Flash メモリに書き込むための POF ファイル生成

3-1. HEX ファイルの生成

2 章で動作した ELF ファイルを HEX ファイルに変換します。

アプリケーション・プロジェクト（例では、hello_world）を右クリックして、Make Targets ⇒ Build を実行します。Make Targets 画面が表示されたら mem_init_generate をハイライトして、Build ボタンをクリックします。



Build ボタンをクリックすると Nios II SBT の Console 画面にログが表示されます。その中で、以下のように表示されている部分が hello_world.elf から ext_flash.hex を作成している部分です。

```
alt-file-convert -I elf32-littlenios2 -O hex --input=hello_world.elf --output=mem_init/ext_flash.hex
                 --base=0x00000000 --end=0x03ffffff --reset=0x00830000 --out-data-width=8
                 --boot="C:/altera/15.1/nios2eds/components/altera_nios2/boot_loader_cfi.srec"
```

入力フォーマット：elf32-littlenios2、出力フォーマット：hex、

入力ファイル：hello_world.elf、出力ファイル：mem_init/ext_flash.hex

ベースアドレス（ext_flash.avl_mem）：0x00000000、エンドアドレス（ext_flash.avl_mem）：0x03FFFFFF

リセットアドレス（Nios II のリセット・ベクタで指定）：0x00830000、出力データ幅：8 ビット

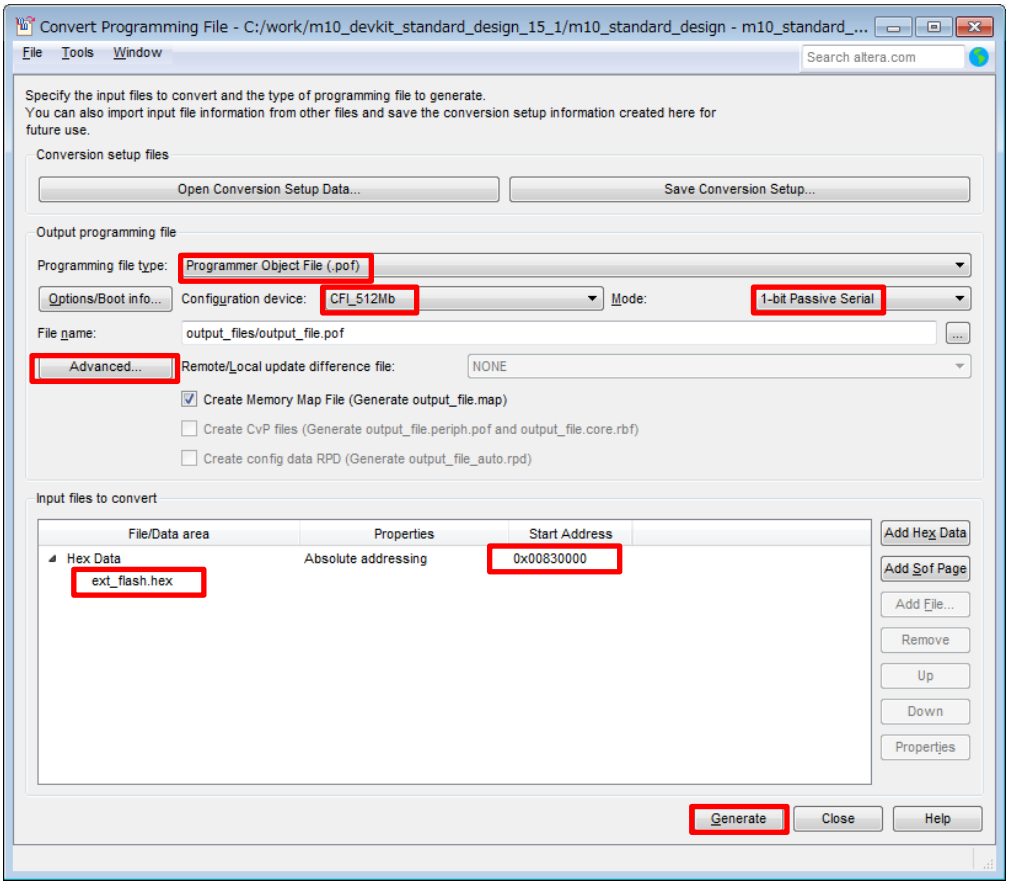
ブートオプション："C:/altera/15.1/nios2eds/components/altera_nios2/boot_loader_cfi.srec"

※ここでのブートオプションは、Quad SPI Flash メモリの 0x83000000 の先頭部分にブートローダを挿入する処理を行っています。Nios II は、起動時にまずこのブートローダ読み込んで、必要な情報を RAM（この例では、外部の DDR3 SDRAM メモリ）に展開します。どの RAM を使用するかは、Nios II SBT の BSP Editor の Linker Script タブでカスタマイズ可能です。

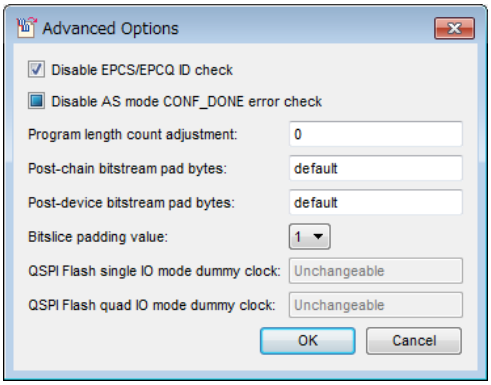
3-2. POF ファイルの生成

作成した HEX ファイルから POF ファイルを生成します。

Quartus Prime に戻って File メニュー ⇒ Convert Programming Files を実行します。画面が起動したら、以下のように設定します。アドレス値 0x00830000 は、ext_flash_hex 内に埋め込まれているので、手動で指定する必要はありません。



Advanced ボタンから Advanced Option を起動して、Disable EPCS/EPCQ ID check をチェックします。



Generate ボタンをクリックし、output_files/output_file.pof ファイルが生成されたことを確認します。

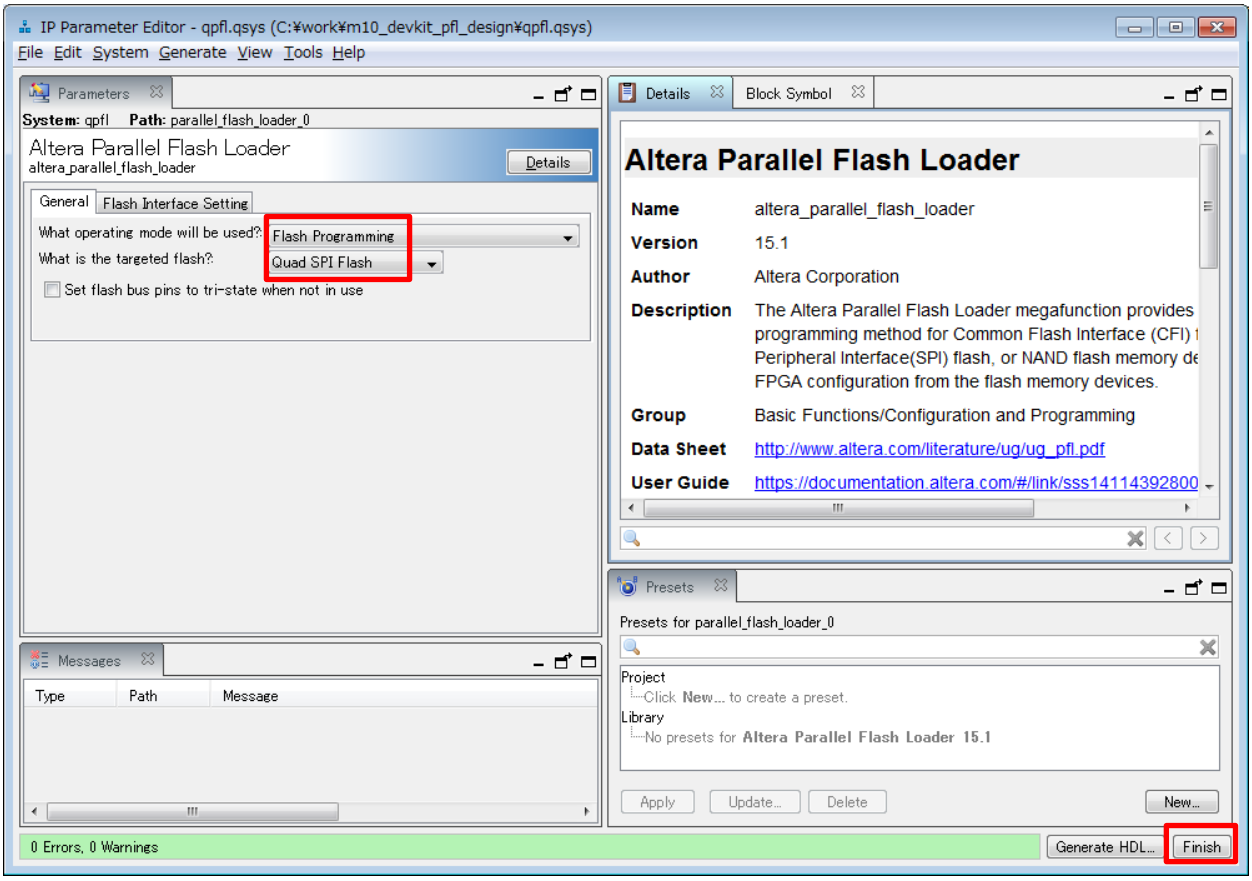
※スタンダード・プロジェクト内には、quartus.ini ファイルがあり、以下のオプションが設定されています。これは、Convert Programming Files で正しい POF ファイルを生成するために必要な設定となります。

PGMIO_SWAP_HEX_BYTE_DATA=ON

4. Quad SPI Flash メモリへの POF ファイルの書き込み

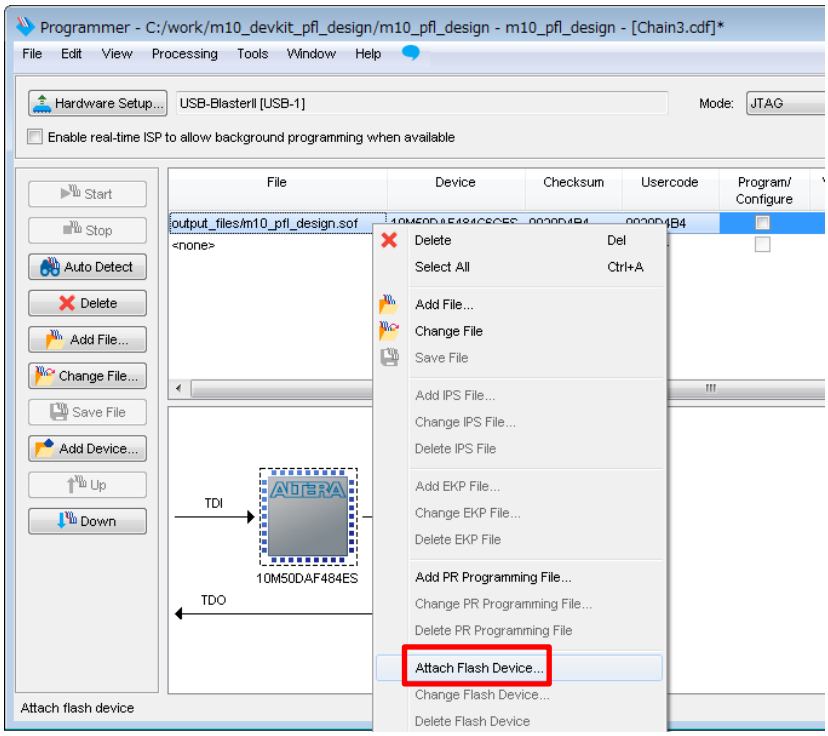
4-1. PFL デザイン

PFL デザインを Quartus Prime で開きます。次に、qpfl.qsys ファイルを Qsys で開きます。Altera Parallel Flash Loader の IP Parameter が起動します。Operation Mode が Flash Programming になっているので、JTAG 経由で Flash メモリ・デバイスのプログラミングを行うモードとなります。Target Flash には Quad SPI Flash が設定されていることを確認します。Finish ボタンで画面を閉じます。

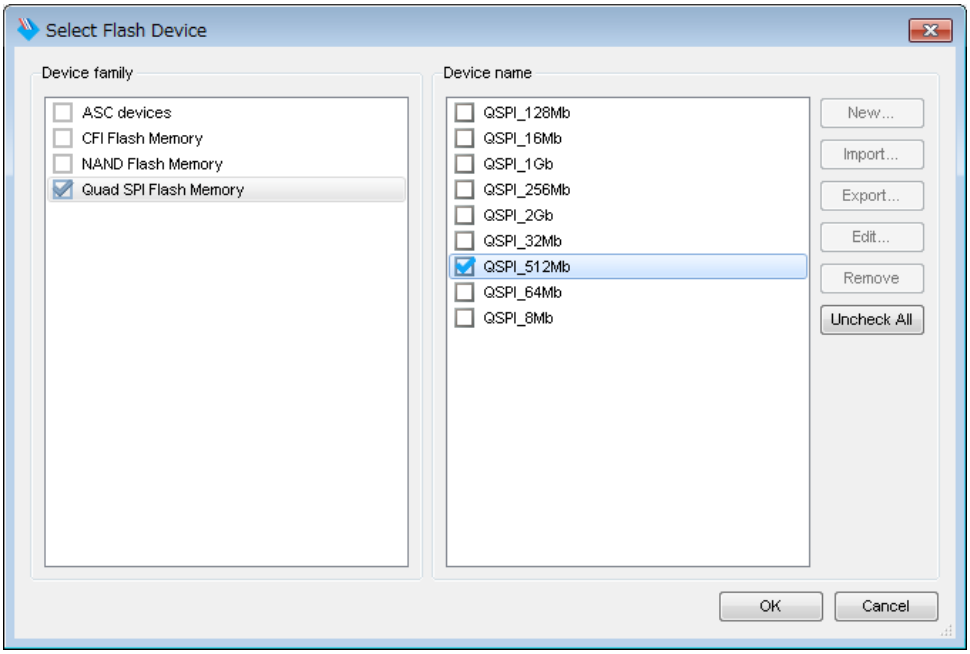


4-2. POF ファイルの Quad SPI Flash メモリへの書き込み

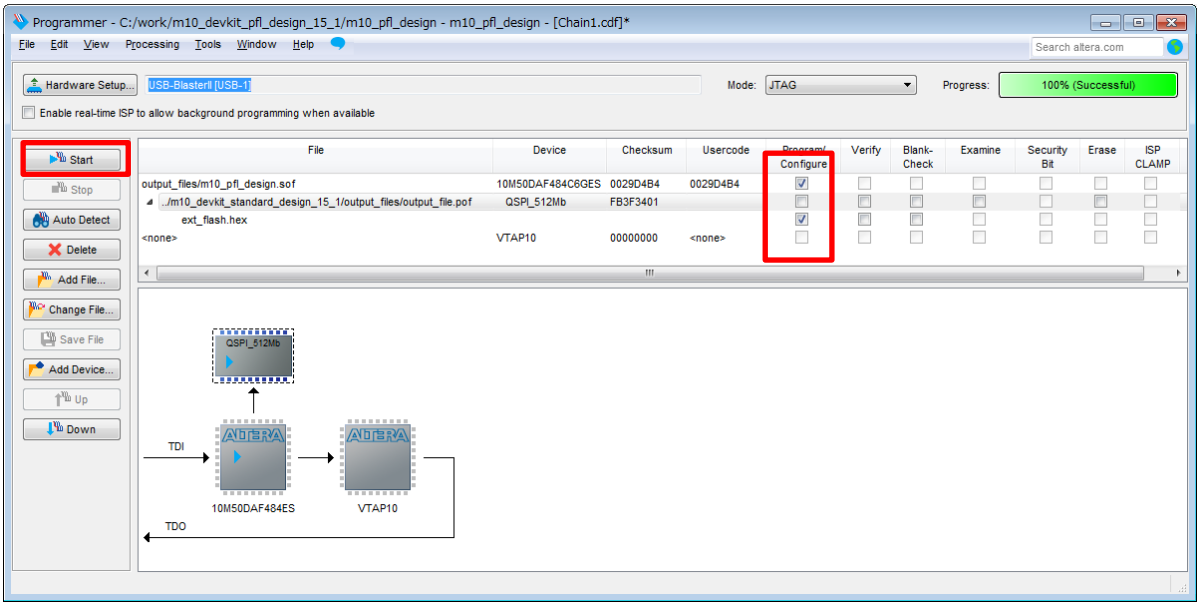
Programmer を起動し、Auto Detect ボタンをクリックします。10M50DA デバイスを指定します。10M50DA デバイスが表示されている行をハイライトし、Change File ボタンから PFL デザインの SOF ファイル（m10_pfl_design.sof）を選択します。次に、右クリックから Attach Flash Device を実行します。



Select Flash Device 画面から、Quad SPI Flash Memoy ⇒ QSPI_512Mb を選択して、OK ボタンをクリックします。



次に、QSPI_512Mb が表示されている行をダブルクリックし、スタンダード・デザイン内で Convert Programming File から作成した POF ファイル（output_files/output_file.pof）を選択します。以下のように、Program/Configure 行部分のチェックをして、Start ボタンで実行します。

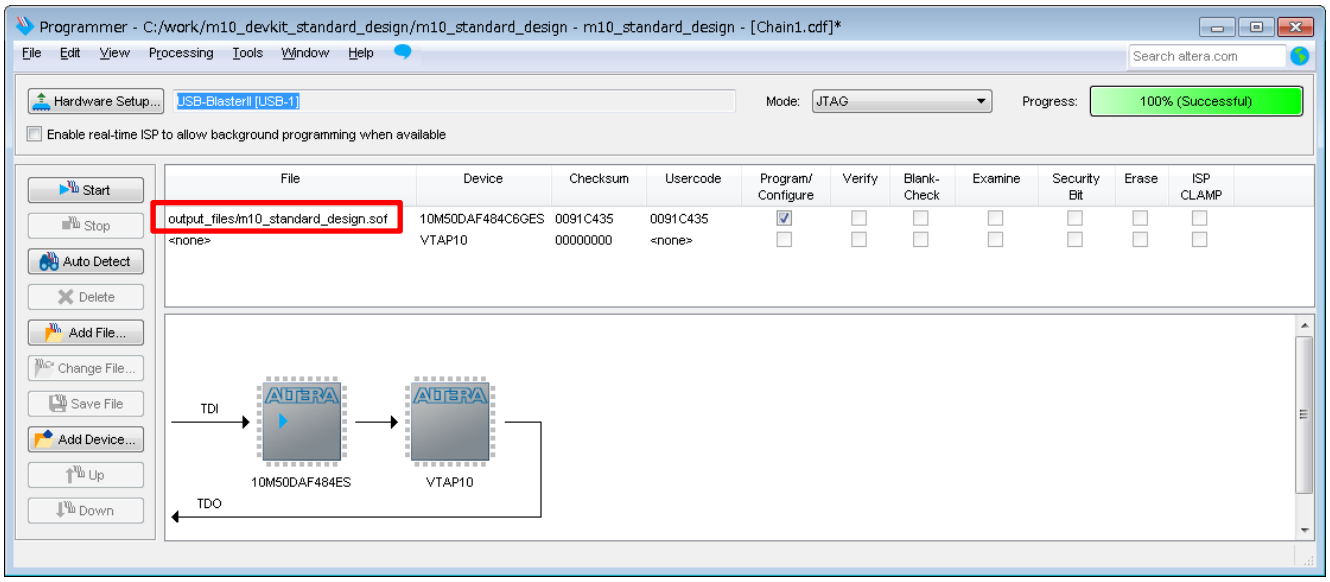


※上記では、ext_flash.hex 行部分のチェックのみ行っていますので、0x00830000 からの領域のみ書き換えを行います。output_file.pof 行にチェックをすると Quad SPI Flash メモリ全体の書き込みを行います（時間がかかります）。

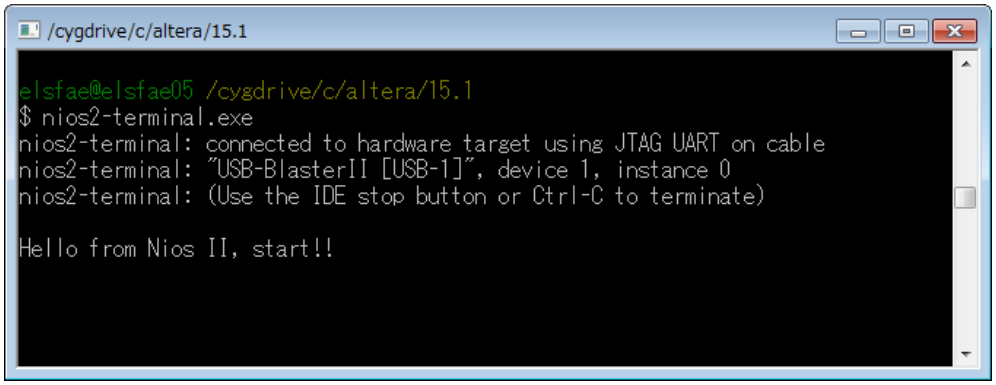
5. スタンダード・デザインの書き込みと動作の確認

5-1. SOF ファイルの書き込みと確認

Quad SPI Flash メモリへの Nios II プログラムの書き込みが正常に行われたかどうかを確認します。スタンダード・デザインの SOF ファイルを Programmer で書き込みます。書き込みが完了した時点でボード上の LED0,1,2,3 が同じタイミングで点滅することを確認します。

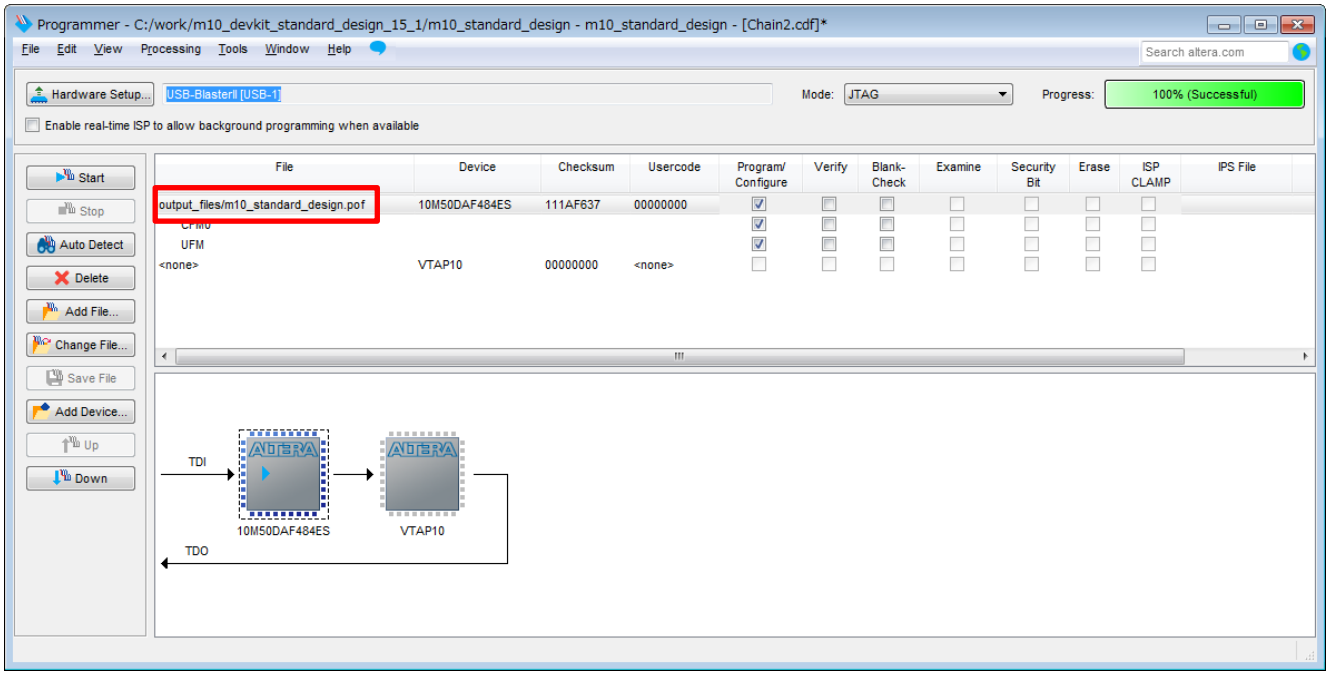


また、Nios II Command Shell を起動し、nios2-terminal.exe を実行することで、JTAG UART からの標準出力が正常に出力されることを確認します。



5-2. POF ファイルの書き込みと確認

次に、スタンダード・デザインの POF ファイル（output_files/m10_standard_design.pof）を MAX 10 デバイスのオンチップ・コンフィギュレーション ROM 領域に書き込みます。



ボードの電源を一旦切り、再度 ON としたときにボード上の LED0,1,2,3 が同じタイミングで点滅することを確認します。また、Nios II Command Shell を起動し、nios2-terminal.exe を実行することで、JTAG UART からの標準出力が正常に出力されることを確認します。

改版履歴

Revision	年月	概要
1	2015 年 12 月	初版