

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

ELSENA

ELSENA,Inc.





Quad SPI Flash メモリからの Nios II ブート方法

<u>目次</u>

1. はじめに
1-1. 環境
1-2. デザイン
1-2-1. スタンダード・デザイン
1-2-2. パラレル・フラッシュ・ローダ・デザイン(PFL デザイン)
1-3. 処理フロー
2. スタンダード・デザインの動作確認5
2-1. Qsys の確認
2-2. Nios II SBT での動作確認
3. Quad SPI Flash メモリに書き込むための POF ファイル生成8
3-1. HEX ファイルの生成
3-2. POF ファイルの生成
4. Quad SPI Flash メモリへの POF ファイルの書き込み10
4-1. PFL デザイン10
4-2. POF ファイルの Quad SPI Flash メモリへの書き込み11
5. スタンダード・デザインの書き込みと動作の確認13
5-1. SOF ファイルの書き込みと確認13
5-2. POF ファイルの書き込みと確認14
改版履歴15

1. <u>はじめに</u>

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 \Rightarrow 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。

Main Vectors Caches and Memory Interfa	aces Arithmetic Instructions MM				
Reset Vector					
Reset vector memory	ext_flash.avl_mem 👻				
Reset vector offset:	0×00830000				
Reset vector:	0×00830000				
* Exception Vector					
Exception vector memory	mem_if_ddr3_emif_0.a 👻				
Exception vector offset:	0×00000120				
Exception vector:	0×08000120				
• East TLD Miss Evention Vester					
Fast TED Miss Exception Vector					
Fast TLB Miss Exception vector memory	none 🚽				
Fast TLB Miss Exception vector offset:	0×00000000				
Fast TLB Miss Exception vector:	0×00000000				

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。

🔌 Programmer - C	:/work/m10_devkit_standard_desigr	ı/m10_standard_desi	ign - m10_st	andard_design	- [Chain1.cdf	:]*						
<u>F</u> ile <u>E</u> dit <u>V</u> iew P	rocessing Tools Window Help 🤇									Search	h altera.com	6
Lardware Setup.	USB-Blasteril (USB-1) P to allow background programming when av	vailable			Mode: JT	ſAG		▼ Pr	rogress:	100%	6 (Successful)	
Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine	Security Bit	Erase	ISP CLAMP	
Stop	output_files/m10_standard_design.sof <none></none>	10M50DAF484C6GES VTAP10	0091C435 00000000	0091C435 <none></none>								
Auto Detect												
Add File												^
Change File												
Add Device	10M50DAF484ES	VTAP10										E
J ^{Wb} Down	↓ TDO											•

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。

Make Targets		×
Make Targets for: hello_world		
Target	Location	Add
mem_init_install		Bamaya
mem_init_generate		Remove
help		Edit
	Build	Cancel

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。

 $alt-file-convert-I\ elf 32-littlenios 2\ -O\ hex\ --input=hello_world.elf\ --output=mem_init/ext_flash.hex$

--base=0x00000000 --end=0x03fffffff --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。

Convert Progra	mming File - C:/wo	k/m10_devkit_standard_o	lesign_15_1/m10_standard_o	design - m10_standard_	
Specify the input file You can also import future use. Conversion setup t	s to convert and the type input file information from	e of programming file to generat n other files and save the conve	t. rsion setup information created he	re for	aitera.com
	Open Conversion S	Setup Data	Sa	ve Conversion Setup	
Output programmin Programming file t <u>u</u> Options/Boot info File <u>n</u> ame: Advanced	g file pe: Programmer Obje Configuration devi output_files/outpu Remote/Local upda V Create Memory Create CvP file Create configuration	ct File (.pof) ce: CFI_512Mb it_file.pof ate difference file: () / Map File (Generate output_file. s (Generate output_file.periph.p data RPD (Generate output_file.	Mode: Mode: Mone map) of and output_file.core.rbf) auto.rpd)	1-bit Passive Se	v rial v v
Input files to conve	/Data area	Properties	Start Address		Add Hey Data
Hex Data	hex	Absolute addressing			Add Sof Page Add Sof Page Add Eile Remove Up Down Properties
				<u>G</u> enerate Close	Help

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. <u>Quad SPI Flash メモリへの POF ファイルの書き込み</u>

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。

IP Parameter Editor - qpfl.qsys (C:¥work¥m10_devkit_pfl_design¥qpfl.qsys) File Edit System Generate View Tools Help			
🙀 Parameters 🛛 🗕 🗗 🗖		🗒 Details 🛛	Block Symbol 🕴 🗕 🗗
System: qpfi Path: parallel_flash_loader_0		L	
Altera Parallel Flash Loader altera.parallel,flash_loader		Altera P	arallel Flash Loader
General Flash Interface Setting		Name	altera_parallel_flash_loader
What operating mode will be used? Flash Programming		Version	15.1
What is the targeted flash : Quad SPI Flash . ▼		Author	Altera Corporation
		Description	The Altera Parallel Flash Loader megafunction provides programming method for Common Flash Interface (CFI) 1 Peripheral Interface(SPI) flash, or NAND flash memory de FPGA configuration from the flash memory devices.
		Group	Basic Functions/Configuration and Programming
		Data Sheet	http://www.altera.com/literature/ug/ug_pfl.pdf
		User Guide	https://documentation.altera.com/#/link/sss14114392800 -
	IL	S	
	Ĵ	🧿 Presets 🛛 🕅	_ d
	F	Presets for paralle	l_flash_loader_0
📲 Messages 🛛 🗕 🗗 🗆	וו	<u></u>	×
Type Path Message		Project Click New to Library No presets for	o create a preset. Altera Parallel Flash Loader 15.1
4	(Apply U	pdate Delete New
0 Errors, 0 Warnings			Generate HDL Finish



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。

🔔 Hardware Setup. 🔲 Enable real-time IS	USB-Blasterll [USB-1] P to allow background programming	when	available		Mo	de: JTAG
▶ [™] Start	File		Device	Checksum	Usercode	Program/ Configure
Stop	output_files/m10_pfl_design.sof <none></none>	*	Delete Select All	DOCODIARIA De	-00000-484 el . rl+A	
X Delete		<u>)</u>))~~	Add File Change File			
Change File	•		Save File Add IPS File Change IPS File			
Add Device			Delete IPS File			
💵 Down		-	Change EKP File Delete EKP File			
	10M50DAF484ES	-	Add PR Programmi Change PR Program Delete PR Program	ing File mming File ming File		
tach flash device			Attach Flash Devic Change Flash Devi	ce		

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。

Select Flash Device		X
Device family	Device name	
 ASC devices CFI Flash Memory NAND Flash Memory Quad SPI Flash Memory 	□ QSPI_128Mlb □ QSPI_16Mlb □ QSPI_256Mlb □ QSPI_20b □ QSPI_32Mlb ☑ QSPI_512Mlb □ QSPI_64Mlb □ QSPI_6Mlb	New Import Export Edit Remove Uncheck All

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。

dware Setup.	USB-Blasteril (USB-1)	ouoilabla			Mode: J	TAG		▼ Pi	rogress:	1009	% (Successf
start	P to allow background programming when	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine	Security Bit	Erase	ISP CLAMP
Log Stop	output_files/m10_standard_design.sof <none></none>	10M50DAF484C6GES VTAP10	0091C435 00000000	0091C435 <none></none>							
Add File iange File Save File id Device		VTAP10									

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。



<u> 改版履歴</u>

Revision	年月	概要
1	2015 年 12 月	初版

免責およびご利用上の注意

弊社より資料を入手されましたお客様におかれましては、下記の使用上の注意を一読いただいた上でご使用ください。

- 1. 本資料は非売品です。許可無く転売することや無断複製することを禁じます。
- 2. 本資料は予告なく変更することがあります。
- 本資料の作成には万全を期していますが、万一ご不明な点や誤り、記載漏れなどお気づきの点がありましたら、本資料を入手されました下記代理店までご一報いただければ幸いです。
 株式会社アルティマ ホームページ: http://www.altima.co.jp
 技術情報サイト EDISON: http://www.altima.jp/members/index.cfm
 株式会社エルセナ ホームページ: http://www.elsena.co.jp
 技術情報サイト ETS : http://www.elsena.co.jp/elspear/members/index.cfm
- 4. 本資料で取り扱っている回路、技術、プログラムに関して運用した結果の影響については、責任を負いかねますのであらかじめご了承ください。
- 5. 本資料は製品を利用する際の補助的な資料です。製品をご使用になる際は、各メーカ発行の英語版の資料もあわせてご利用ください。