

Preloader Generator の使用方法

Ver.14



Preloader Generator の使用方法

<u>目次</u>

1. はじめに	.3
1-1. 必要条件	3
1-2. 関連文書	3
1-3. Preloader とは?	4
1-4. ブート・シーケンス	5
2. Preloader Generator の使用方法	.6
2-1. Embedded Command Shell の起動	6
2-2. bsp-editor(Preloader Generator)の起動	6
2-3. 新規プロジェクトの作成	7
2-4. ハンドオフ・ファイルの指定	7
2-5. Preloader ユーザ・オプションの設定	8
2-5-1. Common オプション設定	9
2-5-2. Advanced オプション設定1	LO
2-6. bsp プロジェクトの生成1	L3
3. Preloader の生成方法	14
3-1. Embedded Command Shell の起動1	۱4
3-2. bsp プロジェクト・ディレクトリへの移動1	۱4
3-3. Preloader のビルド1	۱5
3-4. u-boot のビルド (必要に応じて)1	16
4. Preloader / u-boot の更新方法1	17
4-1. SD カード上イメージの書き換え1	L7
4-2. QSPI Flash メモリの書き換え1	18
改版履歴2	20

1. <u>はじめに</u>

この資料は、Altera SoC Embedded Design Suite (以後、SoC EDS)に付属の Preloader Support Package Generator (別名、bsp-editor)の使用方法について解説します。

1-1. 必要条件

本資料で解説される各種手順を実施するためには次の開発環境があらかじめホスト PC にインストールされ ている必要があります。

- Quartus[®] II v14.0
- SoC EDS v14.0

本資料の解説には v14.0 を使用しており、デフォルトのインストール・パスである以下のロケーションにツー ルをインストールしていることを前提として解説を進めます。本資料内で使用されるスクリーン・キャプチャなど で確認できるツールのインストール・パスなどについては必要に応じて適宜読み替えてご参照ください。

Quartus II のインストール・パス : C:¥altera¥14.0¥quartus

SoCEDS のインストール・パス : C:¥altera¥14.0¥embedded

1-2. 関連文書

本資料は以下の資料および Web サイトの内容をベースに記述されております。これらの情報も合わせてご 確認ください。

- Altera SoC Embedded Design Suite User Guide
- <u>Cyclone V Device Handbook</u>
- RocketBoards.org (ALTERA SoC コミュニティ・ポータルサイト)

1-3. Preloader とは?

Preloader は U-boot second program loader (以後、u-boot spl) をベースに Altera SoC 向けにカスタマイズが 加えられたブートローダです。 Preloader の役割は以下の通りです。

- HPS ピン・マルチプレクスの設定
- HPS IOCSR の設定
- HPS PLL とクロックの設定
- HPS ペリフェラルのリセット解除
- SDRAM の初期化(キャリブレーションなど)
- SDRAM へ次ステージブート・イメージの展開

上記の通り、Preloader は HPS ブロックの初期化と、u-boot や OS を SDRAM にロードする機能を提供しま す。本資料で後述する通り、Preloader は Quartus II / Qsys の設計時に自動生成されるハンドオフ・ファイルを用 いることで自動生成されます。このため、ユーザ側で初期化用ソフトウェアの構築をすることなく Quartus II / Qsys で設定した内容を HPS ブロックに反映することができます。

System Contents 🙁 Address Map 🙁 💆 Parameters 😂	Interconnect Requirements	🛛 Device Family 🕅		
soc_system > hps_0				
Arria V/Cyclone V Hard Processor System	System Contents 🖄 🕯	Address Map 🖾 💆 Parameters 🚿	Interconnect Requirements 💠 Device Family 🌣	
altera_hps	soc_system > hps_0			
FPGA Interfaces Peripheral Pins HPS Clocks SDRAM	Arria V/Cyclone V	Hard Processor System	System Contents 🛞 Address Man 🔅 🗰 P	nameters 🔅 Interconnect Requirements 🔅 Device Family 🔅
General Fnable MPII standby and event signals	altera_rips		and a sector N has 0	
Enable general purpose signals	FPGA Interfaces Periphe	ral Pins HPS Clocks SDRAM		
Enable Debug APB interface	Ethernet Media Acc	ess Controller	Arria V/Cyclone V Hard Processor Sy	stem
Finable System Trace Macrocell hardware events	EMAC0 pin:	Unused		
Fishle EPGA Cross Trigger Interface	EMAC0 mode:	N/A 💌	FPGA Interfaces Peripheral Pins HPS Clocks SD	RAM
Enable EPGA Trace Port Interface Unit	EMAC1 pin:	HPS I/O Set 0 💌	Input Clocks Output Clocks	
Eachie heat from first simple	EMAC1 mode:	RGMI	Clock Sources	
	NAND Flack Contro	ller	Peripheral PLL reference clock source:	EOSC1 clock
I Enable HEGFI Interface	NAND pin	Unused T	SDMMC clock source:	Peripheral NAND SDMMC clock
* AXI Bridges	NAND mode:		NAND clock source	Perinteral NAND SDMMC clock
FPGA-to-HPS interface width: 64-bit 💌		1000	QSPI clock source	Main OSPI check
HPS-to-FPGA interface width: 64-bit 💌	Quad SPI Flash Cor	ntroller	14 MP clock sources	Device and the second s
Lightweight HPS-to-FPGA interface width: 32-bit 💌	Qariph	HPS I/O Set 0 -	14 SP clock course	
FRGAstasHPS SDRAM Interface	QSPI mode:	1 SS 💌	er of cloud source.	Peripheral base clock
Click the 's' and 's' bottom to add and ramma EPGA-to-HPS SDB	SD/MMC Controller		Main PLL Output Clocks - Desired Freque	incles
	SDIO pirs	HPS I/O Set 0 -	Default MPU clock frequency:	800.0 MHz
(2) of rame (2) Avalon-MM Bidirectio 256	SDIO mode:	4-bit Data	✓ Use default MPU clock frequency	
			MPU clock frequency	MHz
	USB Controllers	-	L3 MP clock frequency.	200.0 - MHz
	UCD0 DIN/ interferences	Unused	L3 SP clock frequency:	100.0 • MHz
	USBU PHT Interface mo	0e <u>N/A *</u>	Debug AT clock frequency	25.0 - MHz
+	USB1 pin:	HPS I/O Set 0 💌	Debug clock frequency	12.5 * MHz
	USB1 PHY interface mo	de: SDR with PHY clock output mode 💌	Debug trace clock frequency	25.0 × MHz
▼ Resets	SPI Controllers		L4 MP clock frequency	100.0 MHz
Enable HPS-to-FPGA cold reset output	SPIM0 pin	HPS I/O Set 0 -	L4 SP clock frequency.	100.0 MHz
🕅 Enable HPS warm reset handshake signals	SPIM0 mode:	Single Slave Select *	Configuration/HPS-to-FPGA user 0 clock frequen	CY 1000 MHz
F Enable FPGA-to-HPS debug reset request	SPIM1 pin:	Unused		1
	SPIM1 mode:		Peripheral PLL Output Clocks - Desired F SDMMC clock frequency	requencies MU
	SPIS0 pin:	[linear and	NAND clock frequency	12000 MHZ
	SPIS0 mode	Dirased T	QSPI clock frequency	
	0.000	187.8 ±1	ENACII clock frequency	14000 Mile
			EMAG1 clock frequency	2000 - 1972
			LICD shall be an an an	260.0 V MHz
			USB clock frequency:	200.0 MHz

図 1-1 Qsys HPS の設定例

1-4. ブート・シーケンス

以下に SoC デバイスにおける一般的なブート・シーケンスを示します。HPS のブートは HPS 内部の BootROM を先頭に複数のステージに分かれて実施されます。それぞれのステージは次のステージのブート・イ メージをロードする役割を担っており、これにより順々にブート・イメージがロードされ実行されます。

以下で紹介するブート・シーケンスはあくまでも一例であり、その他の構成で実現することも可能であることに 注意してください。



図 1-2 SoC の一般的なブート・シーケンス

以下に一般的なブート・シーケンスの各ステージにおける役割を示します。

1. BootROM

HPS 内部の ARM プロセッサはリセットが解除されると BootROM にジャンプしソフトウェアの実行を開始します。BootROM では、BSEL ピン、CSEL ピンの設定から Preloader のロードに必要な最低限の初期 化のみを実施し、BSEL ピンで指定された ブート・ソースから Preloader をロードします。BootROM のソ フトウェアの書き換えは不可です。

2. Preloader

前述の通り、Preloader は HPS 部の初期化を実行するためのブートローダです。Preloader を実行する ことで HPS 部の初期化および SDRAM の初期化を実行し、次ステージの Bootloader を SDRAM に展 開し処理を渡します。

3. Bootloader

このステージのブートローダは一般に Operating System (以後 OS) 固有のローダとして配置されます。 ALTERA Linux の場合には、オープンソースで提供される u-boot を Linux のブートローダとして利用して います。

OS / Application
 OS は起動時に必要な処理を実行しアプリケーションの実行を開始します。

ブート・シーケンスに関する詳細は以下の資料をご参照ください。

Booting and Configuration Introduction

2. Preloader Generator の使用方法

このセクションでは、Preloader Generator の使用手順および各種オプションに関して解説します。

2-1. Embedded Command Shell の起動

SoC EDS に付属の Embedded Command Shell を起動します。Windows のスタートメニュー、もしくは、 Windows エクスプローラにて SoC EDS のインストール・フォルダ以下に格納される起動用スクリプトを実行します。

🔗 歴史のプロガラル		📕 embedded					
Altera 14 0 0 200	ミュージック	Color	イスク (C:) + altera + 14.0 + embedded	• • 🐖	embeddedの検索		<u> 1</u>
Initial Unitstall Quartus II Subscription Edit		整理 🔻 🔜 開く 印刷	新しいフォルダー) 1911 -	. 0
Nios Π EDS 14.0.0.200 (64-bit)	JDE1-9-	🚖 お気に入り	名前 🔺	更新日時	和美	サイズ	
Questus II 14.0.0200 (64-bit)	The second se	🎉 ダウンロード	腸 drivers	2014/08/28 11:32	ファイル フォルダー		
	コントロール パネル	■ デスクトップ ③ 鼻にまました場所	🍶 ds-5	2014/08/28 11:43	ファイル フォルダー		
🍶 SoC Embedded Design Suite (EDS) 1		2011年1月1日日の1月1日日の1月1日日の1月1日日の1月1日日の1日日の1日日の1日	鷆 ds-5_installer	2014/08/28 11:34	ファイル フォルダー		
Manuals and Release Notes	デバイスとプリンター 🍡	🧊 ライブラリ	闄 embeddedsw	2014/08/28 11:32	ファイル フォルダー		
Soc EDS 14.0 Command Shell		📑 F#1X2F	🍶 examples	2014/08/28 11:32	ファイル フォルダー		
	既定のブログラム	1000日 ビクチャ	퉬 host_tools	2014/08/28 11:32	ファイル フォルダー		
Standard Soc EDS 14.0		ビナオ	ip 🔒	2014/08/28 11:32	ファイル フォルダー		
📙 Hardware Libs			Embedded_Command_Shellbat	2014/06/19 1:52	Windows バッチ ファイル	2)	KB .
ABM DS-5	→	ユンピューター	embedded_command_shellsh	2014/06/19 1:52	SH ファイル	2 8	ĸВ
· · · · · · · · · · · · · · · · · · ·	000C340 1		📄 envish	2014/06/19 1:52	SH ファイル	6 M	KВ
◆ 前に戻る			version.txt	2014/06/19 21:13	テキスト ドキュメント	13	(B
プログラムとファイルの検索	シャットダウン 🕨	Embedded_Com Windows バッチ つ	imand_Shell.bat 更新日時: 2014/06/ ファイル サイズ: 1.66 KB	'19 1:52 f'Æð	戈日時: 2014/08/28 11:3	3	

図 2-1 Embedded Command Shell の起動

2-2. bsp-editor(Preloader Generator)の起動

Embedded Command Shell に "bsp-editor" とタイプし、bsp-editor の GUI を起動します。



図 2-2 bsp-editor の起動

2-3. 新規プロジェクトの作成

File ⇒ New BSP... を選択し、プロジェクトを新規作成します。

🚠 BSP Editor		<u>_ </u>
File Help		
New BSP Ctrl+N		
Open Ctrl+O		
Save Ctrl+S		
Save As	Varian	
Exit Ctrl+X	VE13011	
)	
Information Problems Processing		
Initializing BSP components		
Finished initializing BSP components. Tota	al time taken = 2 seconds	
Searching for BSP components with categories	gory: os_software_element	
1		

図 2-3 新規プロジェクトの作成

2-4. ハンドオフ・ファイルの指定

Quartus II / Qsys にてプロジェクトをコンパイルした際に自動生成されるハンドオフ・ファイルを指定します。本 資料では SoC EDS に付属のリファレンス・デザイン内に格納されるハンドオフ・ファイルを利用します。

<SoC EDS Install DIR>¥embedded¥examples¥hardware¥cv_soc_devkit_ghrd¥hps_isw_handoff¥soc_system_hps_0

Hardware	
Preloader settings directory:	C:\altera\14.0\embedded\examples\hardware\cv_soc_devkit_ghrd\hps_isw_handoff\soc_system_hps_0
Software	
Operating system:	Preloader Version: default
	I Use default locations
BSP target directory:	C:\altera\14.0\embedded\examples\hardware\cv_soc_devkit_ghrd\software\spl_bsp
BSP Settings File name:	C:\altera\14.0\embedded\examples\hardware\cv_soc_devkit_ghrd\software\spl_bsp\settings.bsp
	☑ Enable Settings File relative paths
	Enable Additional Tcl script
Additional Tcl script:	
	OK L Canad

図 2-4 ハンドオフ・ファイルの指定

Operating System: に Preloader が Version: に default が指定されていることを確認します。BSP target directory にて bsp プロジェクトを生成するロケーションを指定します。

デフォルトでは、"<Quartus II Project DIR>¥software¥spl_bsp"が指定されますが、use default locationsのチェックを外すことで任意のディレクトリを指定することが出来ます。

2-5. Preloader ユーザ・オプションの設定

Preloader に生成時に与えるユーザ・オプションの各種設定を行います。設定には大きく分けて "Common" と "Advanced" があります。それぞれ必要に応じて設定を変更します。

各設定をマウスオーバーすると設定の解説が表示されますので合わせてご確認ください。以降ではそれぞれの設定について解説します。

BSP Editor - C:¥altera¥14.0¥embedded¥ File Help	examples¥hardware¥cv_soc_devkit_ghrd¥sof	tware¥spl_bsp¥settings.bsp		
Main				
SOPC Information file: CPU name: Operating system: Preloader BSP target directory: .\	Version: default			
Settings Spl Spl PRELOADER_TGZ CROSS_COMPILE ⊕-boot ⊡-spl ⊕-reset_assert ⊕-warm_reset_handshake ⊕-boot ⊕-boot ⊕-boot ⊕-boot ⊕-boot ⊕-boot ⊕-performance	spl PRELOADER_TGZ: CROSS_COMPILE: spl.boot BOOT_FROM_QSPI BOOT_FROM_SDMMC BOOT_FROM_NAND BOOT_FROM_RAM QSPI_NEXT_BOOT_IMAGE: SDMMC_NEXT_BOOT_IMAGE: IFAT_SUPPORT FAT_BOOT_PARTITION: FAT_LOAD_PAYLOAD_NAME:	preloader/uboot-socfpga.tar.gz arm-altera-eabi- 0x60000 0x40000 0xc0000 1 1 u-boot.img		
Information Problems Processing	~			
Searching for BSP components with category: driver_element Searching for BSP components with category: software_package_element Added operating system component "spl: 1.0". Generated file "C: \altera\14.0\embedded\examples\hardware\cv_soc_devkit_ghrd\software\spl_bsp\settings.bsp"				
		Generate E <u>x</u> it		



2-5-1. Common オプション設定

Common オプションの設定では、Preloader に関する基本的なオプションに関して設定を行います。以下に設定内容を示します。

大項目	小項目	概要		
spl	PRELOADER_TGZ	Preloader ソース・ファイルのアーカイブを指定します。指定したアーカイブ・ファイルを解		
		凍し利用します。基本的に変更する必要はありません。		
	CROSS_COMPILE	Preloader のビルドに使用するクロスコンパイラを指定します。SoC EDS 付属のコンパイラ		
		が指定されておりますので、基本的に変更する必要はありません。		
spl.boot	BOOT_FROM_QSPI	Preloader に続くブート・イメージを QSPI フラッシュからロードする場合にチェックを入れま		
		す。		
	BOOT_FROM_SDMMC	Preloader に続くブート・イメージを SD カードからロードする場合にチェックを入れます。		
	BOOT_FROM_NAND	Preloader に続くブート・イメージを NAND フラッシュからロードする場合にチェックを入れ		
		ます		
	BOOT_FROM_RAM	Preloader に続くブート・イメージを RAM からロードする場合にチェックを入れます。		
	*上記、BOOT_FROM_XXX の設定			
	QSPI_NEXT_BOOT_IMAGE	BOOT_FROM_QSPI が有効の際、Preloader がロードするブート・イメージが格納されるア		
		ドレスを指定します。		
	SDMMC_NEXT_BOOT_IMAGE	BOOT_FROM_SDMMC が有効の際、Preloader がロードするブート・イメージが格納される		
		アドレスを指定します。		
	NAND_NEXT_BOT_IMAGE	BOOT_FROM_NAND が有効の際、Preloader がロードするブート・イメージが格納されるア		
		ドレスを指定します。		
	FAT_SUPPORT	SD カード内の FAT パーティションに対するアクセスを有効化します。この設定は		
		BOOT_FROM_SDMMC が指定されている時のみ有効です。FAT パーティションに次ステ		
		ージのブート・イメージが格納されている場合に利用します。		
	FAT_BOOT_PARTITION	次ステージのブート・イメージが格納される FAT パーティションの番号を指定します。この		
		設定は、FAT_SUPPORT オプションが指定されている時のみ有効です。		
	FAT_LOAD_PAYLOAD_NAME	次ステージのブート・イメージのファイル名を指定します。この設定は、FAT_SUPPORT オプ		
		ションが指定されている時のみ有効です。		

表 2-1 Common オプション設定一覧

2-5-2. Advanced オプション設定

Advanced オプション設定では、Preloader に関する高度な機能について設定します。基本的に変更の必要はありませんが状況に応じて設定を変更してください。

spl.reset_assert	
L4WD1	
C OSC1TIMER1	
SPTIMER1	
☐ GPI00	
GPI01	
GPI02	
🗖 DMA	
SDR	
spl.warm_reset_handshake	
FPGA	
₩ ETR	
SDRAM	
spl.boot	
WATCHDOG ENABLE	
FPGA_MAX_SIZE:	0x10000
FPGA_DATA_BASE:	0xffff0000
FPGA_DATA_MAX_SIZE:	0x10000
STATE_REG_ENABLE	
BOOTROM_HANDSHAKE_CFGIO	
WARMRST_SKIP_CFGIO	
SDRAM_SCRUBBING	
SDRAM_SCRUB_BOOT_REGION_START:	0x1000000
SDRAM_SCRUB_BOOT_REGION_END:	0x2000000
SDRAM_SCRUB_REMAIN_REGION	
spl.debug	
DEBUG_MEMORY_WRITE	
DEBUG_MEMORY_ADDR:	0xffffd00
DEBUG_MEMORY_SIZE:	0x200
HARDWARE_DIAGNOSTIC	
SKIP_SDRAM	
spl.performance	
SERIAL_SUPPORT	

図 2-6 Advenced オプション設定



表 2-2 Advanced オプション設定一覧

大項目	小項目	概要
spl.reset_assert		Preloader 実行時に該当ペリフェラルのリセットの解除を実行
		するか指定します。チェックを入れるとリセット状態のままとな
		ります。
spl.warm_reset_handshake		WarmReset 時に該当ペリフェラルとのハンドシェイク機能の
		使用有無を指定します。
spl.boot	WATCHDOG_ENABLE	ウォッチドッグ・タイマの使用有無を指定します。各種デバッグ
		時には無効化することを推奨します。
	CHECKSUM_NEXT_IMAGE	次ステージのブート・イメージのロード時にチェックサムによる
		エラーチェックの実施有無を指定します。
	EXE_ON_FPGA	Preloader を FPGA メモリ上で実行する再にチェックを有効にし
		ます。このオプションは、BSEL を FPGA に指定した場合に利
		用します。
	FPGA_MAX_SIZE	FPGA メモリに配置可能な.text、rodata の最大サイズを指定
		します。Preloader のビルド時にこのサイズよりも実際のコー
		ドサイズが大きい場合にビルドエラーを返します。
	FPGA_DATA_BASE	EXE_ON_FPGA オプション有効時に、.data、.bss、malloc、stack
		を配置するベースアドレスを指定します。
	FPGA_DATA_MAX_SIZE	Preloader のビルド時にこのサイズよりも実際のデータ
		(.data 等)が大きい場合にビルドエラーを返します。
	STATE_REG_ENABLE	Preloader 実行時に STATE R レジスタに STATE_VALID データ
		を書き込みます。これは BootROM に対して Preloader が正
		常にロードされたことを示します。
	BOOTROM_HANDSHAKE_CFGIO	IOCSR および Pin MUX の初期化時に BootROM とのハンド
		シェイクを実行します。本設定が有効時に WarmReset が発
		生した場合、Preloader が初期化を実施していても BootROM
		にて再設定を実行します。
	WARMRST_SKIP_CFGIO	本設定が有効な場合、Preloader は IOCSR および Pin MUX
		の 設 定 を ス キ ップ し ま す 。こ の 設 定 は 、
		BOOTROM_HANDSHAKE_CFGIO が無効な場合にのみ有効と
		なります。
	SDRAM_SCRUBBING	SDRAM ECC 有効時にメモリの初期化を実行します。
	SDRAM_SCRUB_BOOT_REGION_START	SCRUB を実行するメモリの先頭アドレスを指定します。
	SDRAM_SCRUB_BOOT_REGION_END	SCRUB を実行するメモリの終了アドレスを指定します。
	SDRAM_SCRUB_REMAIN_REGION	次ステージのブート・イメージをロード中に上記 SCRUB オプ
		ションで指定されない残りのメモリ領域の初期化を実行する
		か指定します。
spl.debug	DEBUG_MEMORY_WRITE	デバッグ情報をメモリに書き出すオプションの使用有無を指
		定します。UART が使用出来ないシステムに有効です。

ALTIMA DEBUG MEMORY ADDR デバッグ情報を書き出すメモリの先頭アドレスを指定します。 DEBUG_MEMORY_SIZE デバッグ情報を書き出すメモリに割り当てるメモリサイズを指 定します。 SEMIHOSTING セミホスティング機能の使用有無を指定します。このオプショ ンはデバッガと接続されていることが前提のオプションとなる ため注意してください。UART が使用出来ないシステムにおけ る Preloader のデバッグに有効です。 HARDWARE_DIAGNOSTIC SDRAM メモリの簡易テストを実行します。このオプションを利 用するためには少なくとも 1GB のメモリ領域が必要です。 SKIP_SDRAM Preloader の実行時に SDRAM メモリの初期化およびキャリ ブレーションをスキップします。 SERIAL_SUPPORT UART を使用したログ出力を有効化します。 spl.performance

2-6. bsp プロジェクトの生成

各種設定が完了後、bsp プロジェクトを生成します。生成される bsp プロジェクトには *.c 、 *.h 、 Makefile を含む Preloader を生成(ビルド)するために必要なファイル群がすべて生成されます。これらのファイルは、 「<u>2-4 ハンドオフ・ファイルの指定</u>」で "BSP target directory" に指定したロケーションに出力されます。

bsp-editor 右下の "Generate" ボタンを押下しプロジェクトを生成します。本資料では、以下のディレクトリが 該当します。

<SoC EDS Install DIR>¥embedded¥examples¥hardware¥cv_soc_devkit_ghrd¥software¥spl_bsp



図 2-7 bsp プロジェクトの生成

3. Preloader の生成方法

このセクションでは、前述の手順で生成した bsp プロジェクトを利用し Preloader を生成する手順について解説します。

3-1. Embedded Command Shell の起動

Embedded Command Shell を起動します。起動方法については前述の「<u>2-1 Embedded Command Shell の起</u> <u>動</u>」を参照ください。

3-2. bsp プロジェクト・ディレクトリへの移動

cd コマンドを利用して Preloader Generator (bsp-editor) で生成した bsp プロジェクトのルートディレクトリに移動します。本資料の場合は以下のディレクトリが該当します。

<SoC EDS Install DIR>¥embedded¥examples¥hardware¥cv_soc_devkit_ghrd¥software¥spl_bsp



図 3-1 bsp プロジェクト・ディレクトリへの移動

3-3. Preloader のビルド

"make all" コマンドを実行し Preloader を生成します。



図 3-2 Preloader のビルド

実行後、"<Quartus II Project Top DIR>¥software¥spl_bsp" 以下に、preloader-mkpimage.bin という名称のバイナ リ・ファイルが生成されていることを確認します。このファイルは BootROM にて参照される Preloader 用のヘッ ダ情報を付加したバイナリとなっています。SD カードおよび QSPI フラッシュメモリ等への書き込みはこのバイ ナリ・ファイルを利用します。

理 👻 📄 開く 新しい	フォルダー			1
- お気に入り -	名前一	更新日時	種類	14
🚺 ダウンロード	📕 generated	2014/08/28 18:29	ファイル フォルダー	
📰 デスクトップ	🌗 uboot-socfpga	2014/08/28 18:31	ファイル フォルダー	
📃 最近表示した場所	📄 Makefile	2014/08/28 18:29	ファイル	
	📄 preloader.ds	2014/08/28 18:29	DS ファイル	
ライブラリ	📄 preloader-mkpimagebin	2014/08/28 18:33	BIN ファイル	
■ ドギュメント	settingsbsp	2014/08/28 18:29	BSP ファイル	
► C0+9	🗋 uboot.ds	2014/08/28 18:29	DS ファイル	
□ しノカ 小 ミュージッカ				
• • • • • • • •				
リコンピューター				
_				
ネットワーク	•		1	

図 3-3 生成された Preloader バイナリ・ファイル

【注記】

本資料では、Windows® 7 Professional を使用して動作の確認を行っております。

ホスト PC の OS が Windows[®] 10 の場合、Preloader の生成でエラーが発生する場合が確認されてお ります。

もしご使用の OS が Windows[®] 10 でエラーが発生する場合は、下記のアルティマ技術サポートのコンテ ンツページで紹介している、bsp プロジェクト内 Makefile の編集が必要となりますのでご注意ください。

Windows® 10 における Preloader のビルドエラー

3-4. u-boot のビルド (必要に応じて)

"make uboot" コマンドを実行し u-boot を生成します。



図 3-4 u-boot の生成

上記コマンドを実行すると、以下のディレクトリに Preloader にてロード時に参照されるヘッダを付加した u-boot バイナリ・イメージである u-boot.img が生成されます。

<Quartus II Project Top DIR>¥software¥spl_bsp¥uboot-socfpga¥u-boot.img

	shipsh , apost sectings .		aune 🛪
里 👻 🤠 ディスクイメージ	の書き込み 🔻 新しいフォルダー) 🗄 🔸 🛄
- お与(r 入 n 🌙	名前 🔺	更新日時	種類
	Makefile.orig	2014/06/19 22:15	URIG 7717
<u>■ デフカトッ</u> ↑	📄 mkconfig	2014/06/19 22:15	ファイル
■ 1 (1) (1) (1) (1) (1) (1) (1) (1) (1) (📄 README	2014/06/19 22:17	ファイル
2012-2013-012-2011	📄 rules.mk	2014/08/28 18:31	MK ファイル
ライブラリ	📄 snapshot.commit	2014/06/19 22:17	COMMIT ファイル
1 K+1X21	📄 System.map	2014/08/28 18:39	MAP ファイル
ピクチャ	📄 u-boot	2014/08/28 18:39	ファイル
📑 ೮೯೫	📄 u-bootbin	2014/08/28 18:39	BIN ファイル
👌 ೩೨ – ೮୬೮	💽 u-bootime	2014/08/28 18:39	ディスク イメージ ファ
	u-boot.lds	2014/08/28 18:39	LDS ファイル
コンピューター	📄 u-boot.map	2014/08/28 18:39	MAP ファイル
	📄 u-boot.srec	2014/08/28 18:39	SREC ファイル
- AMPJ-0	1		

図 3-5 生成された u-boot イメージ・ファイル

4. Preloader / u-boot の更新方法

このセクションでは前述の手順で新規に作成した Preloader バイナリ・ファイルをセットアップ済みの SD カードおよびオンボード上の QSPI Flash メモリに書き込む方法について解説します。本資料で紹介する手順はあくまでも一例であり、他の方法で書き換えることも可能です。

4-1. SD カード上イメージの書き換え

SD カード上イメージの書き換えは、RocketBoards.org に公開されている SD カードイメージ、もしくは SoC EDS 付属の SD カードイメージが書き込まれていることを前提として解説を進めます。

SD カードのセットアップが完了していない場合には、「<u>1-2</u>関連文書」で紹介した<u>Altera SoC Embedded</u> <u>Design Suite User Guide</u>のGetting Started Guides を参考にセットアップを進めてください。なお、SoC EDS に付属 のSD カードイメージは以下に保存されています。

<SoC EDS Install DIR>¥embedded¥embeddedsw¥socfpga¥prebuilt_images¥sd_card_linux_boot_image.tar.gz

SoC EDS v14.0 には Windows マシンから直接 SD カードの中身を書き換えるためのツール (ALTERA Boot Disk Utility) が同梱されており、上記いずれかの SD カードイメージでセットアップ済みの SD カードに対し、 Preloader / u-boot のデータのみを部分的に書き換えることが可能です。

ALTERA Boot Disk Utility を実行する際は、Embedded Command Shell を管理者権限で起動する必要がありますので注意してください。

Preloader / u-boot を書き換える場合には以下のコマンドを使用します。

- Preloader: \$ alt-boot-disk-util -p preloader-mkpimage.bin -a write -d D
- u-boot: \$ alt-boot-disk-util -b uboot-socfpga/u-boot.img -a write -d D

* 上記コマンドの -d オプションで指定するドライブレターはホスト PC 環境により異なります。お使いの環境 に合わせて読み替えてください。

💽/cygdrive/c/altera/14.0/embedded/examples/hardware/cv_soc_devkit_ghrd/software/spl_bsp 📃 🔲 🗙	💽/cygdrive/c/altera/14.0/embedded/examples/hardware/cy_soc_devkit_ghrd/software/spl_bsp 📰 🔲 🗙
Altera Embedded Command Shell	Altera Embedded Command Shell
Version 14.0	Version 14.0
altima@altima-PC~~ \$ cd "C:¥altera¥14.0¥embedded¥examples¥hardware¥cv_soc_devkit_ghrd¥software¥spl _bsp"	altima@altima-PC ~~ \$ cd "C:¥altera¥14.0¥embedded¥examples¥hardware¥cv_soc_devkit_shrd¥software¥spl _bsp"
altima®altima-PC /cygdrive/c/altera/14.0/embedded/examples/hardware/cv_soc_devki t_shrd/software/spl_bsp \$ alti-boot-disk-util -p preloader-mkpimage.bin -a write -d D Altera Boot Disk Utility Copyright (C) 1991-2014 Altera Corporation	altime®altima-PC /cygdrive/c/altera/14.0/embedded/examples/hardware/cy_soc_devki t_ghrd/software/spl_bsp \$ alt-boot-disk-util -b uboot-socfpga/u-boot.img -a write -d D Altera Boot Disk Utility Copyright (C) 1991-2014 Altera Corporation
Altera Boot Disk Utility was successful.	Altera Boot Disk Utility was successful.
altima®altima-PC /cygdrive/c/altera/14.0/embedded/examples/hardware/cv_soc_devki t_shrd/software/spl_bsp \$_ 	altima@altima=PC /cvgdrive/c/altera/14.0/embedded/examples/hardware/cv_soc_devki t_ghrd/software/spl_bsp \$_

図 4-1 SD カードイメージの書き換え

Linux マシンであれば dd ユーティリティを使用することで同様の処理を実現可能です。Linux マシンにおける 書き換え方法の詳細は以下のリンクをご参照ください。

GSRD v14.0 - SD Card

Updating Individual Elements on the SD card

4-2. QSPI Flash メモリの書き換え

オンボード上の QSPI Flash メモリの場合、SoC EDS に付属の HPS Flash Programmer が使用出来ます。この ツールは JTAG を介し QSPI Flash メモリに直接書き込みを実行します。



図 4-2 HPS Flash Programmer による Flash 書き込み

作業の開始前に BSEL ピンが QPSI ブートを選択していることを確認してください。

BOOTSEL Field Value	Flash Device
0x0	Reserved
0x1	FPGA (HPS-to-FPGA bridge)
0x2	1.8 V NAND flash memory
0x3	3.3 V NAND flash memory
0x4	1.8 V SD/MMC flash memory with external transceiver
0x5	3.3 V SD/MMC flash memory with internal transceiver
0x6	1.8 V SPI or quad SPI flash memory
0x7	3.3 V SPI or quad SPI flash memory

図 4-3 BSEL ピンの設定

HPS Flash Programmer にて Preloader / u-boot を書き換えるコマンドは以下の通りです。

- quartus_hps -c 1 -o p preloader-mkpimage.bin
- quartus_hps -c 1 -o p -a 0x60000 uboot-socfpga/u-boot.img



図 4-4 HPS Flash Programmer を利用したイメージの書き込み

HPS Flash Programmer に関する詳細は、「<u>1-2 関連文書</u>」で紹介した <u>Altera SoC Embedded Design Suite User</u> <u>Guide</u> を参照ください。

<u> 改版履歴</u>

Revision	年月	概要
1	2014 年 9 月	新規作成
2	2018 年 9 月	 書式変更 Windows[®] 10 使用の際の Preloader 生成における注記を追加 リンク URL 修正
3	2020年1月	 15ページ【注記】を修正 20ページ「免責およびご利用上の注意」内のリンク修正

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。
 株式会社マクニカ 半導体事業 お問い合わせフォーム
- 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。