

# Nios II - オンチップ・メモリ からのブート手順

ver.14



2014年9月 Rev.2

ELSENA,Inc.





# Nios II - オンチップ・メモリからのブート手順

# <u>目次</u>

1. はじめに	3
1-1. ブート用オンチップ・メモリの設定	3
1-2. Nios Ⅱ プロセッサのリセット・ベクタと例外ベクタの設定	4
2. Nios II SBT を使用したときのオンチップ・メモリからのブート手順	5
2-1. BSP Editor の設定 (ROM 設定のオンチップメモリからブートする場合)	5
2-2. HEX ファイルの生成方法	6
2-3. Quartus II での設定とコンパイル	9
2-4. HEX ファイル更新時の Quartus II プロジェクトへの反映方法	11
改版履歴	13

# 🔥 ALTIMA

## 1. <u>はじめに</u>

この資料は、Nios<sup>®</sup> II のソフトウェア・プログラムを FPGA デバイス内のオンチップ・メモリからブートする際の 設定方法について説明します。前提として、Quartus<sup>®</sup> II (本例では、ver 14.0を使用)にてハードウェア・デザインが 完成していて、Nios II Software Build Tool for Eclipse (以下、Nios II SBT) から JTAG 経由でアプリケーショ ン・ソフトウェアが正常に動作することが必要となります。

#### 1-1. ブート用オンチップ・メモリの設定

ソフトウェア・プログラムを配置するオンチップ・メモリをシステムに追加しますが、 ROM/RAM のどちらかを選択 する必要があります。以下の特徴を参考に、用途によって選択してください。

- RAM の場合: RAM 領域としても使用可能のため、別途 RAM を用意する必要がありません。
   ただし、ソフトウェアの作りによってはソフトウェア・プログラム本体を破壊してしまい
   起動できなくなる恐れがあります。
- ・ ROM の場合: RAM 領域が別途必要です。 (この資料では、RAM 領域としてオンチップ・メモリを使用した例を紹介します。)

🚈 Parameters 🛛		- 5	i 🗆
DE2_115_QSYS > onchip_ram			
On-Chip Memory (RAM or ROM) altera_avalon_onchip_memory2			
▼ Memory type			
Type:	RAM (Writable) 🔻		
🗖 Dual-port access			
Single clock operation			
Read During Write Mode:	DONT_CARE 💌		
Block type:			
▼ Size			
Data width:	32 💌		
Total memory size:	131072	bytes	
🗖 Minimize memory block usage (may imp	act fmax)		
▼ Read latency			
,			

#### 1-2. Nios II プロセッサのリセット・ベクタと例外ベクタの設定

Qsys 上にある Nios II のリセット・ベクタの設定をオンチップ・メモリに設定する必要があります(下図参照)。 この設定で、FPGA 起動後に Nios II がリセット・ベクタで指定されたアドレスからソフトウェア・プログラムをロード しブートが開始します。

ブート用オンチップ・メモリが ROM の場合、例外・ベクタの設定を RAM 領域に設定する必要があります。今回の例では、オンチップ・メモリを追加します。

	1			+ _
			- 1	
DE2_115_QSYS > cpu				
Nios II Processor				
altera_inosz_qsys				
Core Nios II Caches and Memory	/Interfaces Advanced Featur	res MMU and MPU Settings	JTAG Debug Module	
Select a Nios II Core			· · · · · · · · · · · · · · · · · · ·	
Nios II Core:	⊙ Nios II/e			
	Nios II/s			
	⊂ Nios II/f			
	Nios II/e	Nios II/s	Nios II/f	
Nios II Selector Guide	RISC 32-bit	RISC 32-bit Instruction Cache Branch Prediction Hardware Multiply Hardware Divide	RISC 32-bit Instruction Cache Branch Prediction Hardware Multiply Hardware Divide Barrel Shifter Data Cache Dynamic Branch Prediction	
Memory Usage (e.g Stratix IV)	Two M9Ks (or equiv.)	Two M9Ks + cache	Three M9Ks + cache	
Hardware Arithmetic Oper Hardware multiplication type:	ration			
T Hardware divide	Empedded Marciple			
- Prost Venter				
Reset vector memory.	onchin mmst			
Reset vector offset:				
Reset vector:	0x08240000			
	,			
Exception vector Exception vector memory.	onchin ramet			
Exception vector offset:				
Exception vector:	0.0820000			
	10//02/00000			
MMU and MPU				
Include IMMU				
Only include the MMU using an i	operating system that explicit	y supports an MMU.		
Fast TLB Miss Exception vector	r memory: none	<b>V</b>		
Fast TLB Miss Exception vector	r offset: 0.0000000			
Fast TLB Miss Exception vector	r:  0x0000000			



### 2. Nios II SBT を使用したときのオンチップ・メモリからのブート手順

#### 2-1. BSP Editor の設定 (ROM 設定のオンチップ・メモリからブートする場合)

Nios II SBT にて BSP・ソフトウェア・プロジェクト(本例では、soft\_test\_bsp)を右クリックして、プルダウン・メニューから BSP Editor を起動します。そこで、Main タブで下図のように hal.linker の設定をします。

👬 Nios II BSP Editor – settings.bsp			
File Edit Tools Help			
Main Software Packages Drivers Linker Script Enable File	Seneration Target BSP Directory		
SOPC Information file:\\DE2_115_QSYS.sopcinfo CPU name: cpu Operating system: Altera HAL BSP target directory: C:\workshop\test\DE2_115_golden_	Version: default 💽		
⊡-Settings ⊕-Common ⊕-Advanced	hal.linker         ✓ allow_code_at_reset         ● enable_alt_load         ● enable_alt_load_copy_rodata         ● enable_alt_load_copy_exceptions         hal.make         build_pre_process:         ar_pre_process:         bsp_cflags_defined_symbols:         ar_post_process:         as:	none none none none none none	
Information Problems Processing			
Mapped module: "top" to use the default driver version.     Mapped module: "timer" to use the default driver version.     Mapped module: "timer" to use the default driver version.     Mapped module: "bag'to use the default driver version.     Mapped module: "systd" to use the default driver version.     Mapped module: "systd" to use the default driver version.     Mapped module: "systd" to use the default driver version.     Mapped module: "systd" to use the default driver version.     Mapped module: "systd" to use the default driver version.     Mapped module: "systd" to use the default driver version.     Mapped module: "systd" to use the default driver version.     Mapped module: Top System Strong File.     Joang SSP settings Fine settings File.     Pinished loading SOPC Builder system info file ".\\DE2_115	1. _QSYS.sopcinfo [relative to settings file]"		
			Generate Exit

Linker Script タブで、.entry 領域以外は全て RAM 領域のオンチップ・メモリに設定する。

le Edit Tools I	Help							
1ain Software Pac	kages Driver	s Linker Script	Enable File Gene	ration	Target BSP Dir	ectory		
inker Section Mapp.	bings						-	
Linker Section I	Name 🛆	Linker Regi	on Name	M	lemory Device I	Name	A	3d
.bss		onchip_ram		onchip_	_ram	<b></b>	Rem	10Ve
entry		reset		onchip_	_rom		Restore	Defaults
exceptions		onchip_ram		onchip_	_ram			
heap		onchip_ram		onchip_	ram			
rodata		onchip_ram		onchip_	_ram			
rwoata		onchip_ram		onchip_	_ram			
stack		onchip_ram		onch i p_	_ram	•		
nker Memory Rea	ions							
inker Region Nam	e Addre:	ss Range 🔻	Memory Devic	e Name	Size (bytes)	Offset (bytes)	[ A	d
nchip_rom	0×0824002	0 - 0×0825FFFF	onchip_rom		131040	32	Rem	10Ve
eset	0×0824000	0 - 0×0824001F	onchip_rom		32	0	Restore	Defaults
nchip_ram	0×0820000	0 - 0×0821FFFF	onchip_ram		181072	0		
ram	0×0800000	0 - 0×081FFFFF	sram		2097152	0		
dram	0×0000000	0 - 0×07FFFFFF	sdram		134217728	0	Add Memo	bry Device
							Remove Me	mory Device
							Memory	/ Usage
aved out entries	are automa	tically created at	nonorato timo	They	are not editab	la or narejeta	1 in the BSP cet	tings file
ayou out entites	are autorna	icany createu at	generate time	. iney a	are not euitab	ne or persister	a in the bor set	ango me.
formation Proble	ms Processin	ng						
Mapped module:	"cpu" to use t	he default driver v	ersion.					
Mapped module:	"timer" to use	the default driver	version.					-
Mapped module:	"ledr" to use t	he default driver v	ersion.					Ĩ

#### 2-2. HEX ファイルの生成方法

Nios II SBT にてビルド済のアプリケーション・ソフトウェア・プロジェクト(本例では、soft\_test) を右クリックして、 プルダウン・メニューから Make Targets  $\Rightarrow$  Build を選択します。

Project E	kplore	er X 🗖 🗖 🖟	led_output.c 🛛	
		🖻 🔄 💱 🏹	<pre>#include <stdio.h></stdio.h></pre>	
<ul> <li>✓ soft_t</li> <li>&gt; Similar     <li>&gt; Similar     <li>&gt; Observe     <li>&gt; So     <li>&gt; Constraints     <li>&gt; So     <li>&gt; Constraints     <li>&gt; Main     <li>&gt; Text     </li> <li>&gt; <ptext< p=""> </ptext<></li> <li>&gt; <ptext< p=""> <th></th><th>New Go Into Open in New Window Copy Paste Delete Remove from Context Source</th><th>#include <io.h>  Ctrl+C Ctrl+V Delete Ctrl+Alt+Shift+Down</io.h></th><th><pre>I, start!!"); 0x55);</pre></th></ptext<></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></ul>		New Go Into Open in New Window Copy Paste Delete Remove from Context Source	#include <io.h>  Ctrl+C Ctrl+V Delete Ctrl+Alt+Shift+Down</io.h>	<pre>I, start!!"); 0x55);</pre>
iii so iii so ⊳ ﷺ soft_t	1 23 23	Move Rename Import Export	F2	Properties Nios II Console 33 USB-Blaster on localhost [USB-0] device ID: 1 instance
	8	Build Project Clean Project Refresh Close Project Close Unrelated Projects Build Configurations	F5	
		Make Targets		Create
		Index	,	Build Shift+F9
		Show in Remote Systems Convert To Run As	view ,	Kebullo Läst Tärget F9

下図のように、Make Targets 画面が起動したら、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: soft_test	×
Target	Location Add
init install	Remove
mem_init_generate	Keniove
le help	Edit
•	Þ
Build	Cancel

この操作によって、mem\_init フォルダがアプリケーション・プロジェクトの中に生成されます。



下図のように、Windows エクスプローラーから見ても mem\_init フォルダが生成されていることが確認できます。

	i. D. h	is set 14.0 software soft both				×
	lios2_bas	ic_prj_14_0 ▶ software ▶ soft_test ▶	▼ + <sub>2</sub>	soft_test())検索		Q
整理 ▼ 🗦 開く 🗦	ライブラリ	ルに追加 ▼ 共有 ▼ 書き込む 新し	いフォルダー		•== -	0
☆ お気に入り	Â	名前	更新日時	種類	サイズ	
] ダウンロード		퉬 mem_init	2014/07/01 11:05	ファイル フォル…		
📃 デスクトップ		🎍 obj	2014/07/01 11:06	ファイル フォル…		
1週 最近表示した場所		cproject	2014/07/01 10:06	CPROJECT ファ	36 KB	
	=	.project	2014/07/01 10:06	PROJECT ファイル	3 KB	
		create-this-app	2014/07/01 10:06	ファイル	4 KB	
		C led_output.c	2014/07/01 10:06	C言語ソースファ	1 KB	
■ ドキュメント		Makefile	2014/07/01 10:06	ファイル	35 KB	
📔 ピクチャ		nios2_system_epcq_boot_rom.flash	2014/07/01 11:06	FLASH ファイル	149 KB	
📑 ビデオ		🖳 readme.txt	2014/07/01 10:06	テキスト文書	1 KB	
⊿ ミュージック		soft_test.elf	2014/07/01 11:06	ELF ファイル	760 KB	
-		soft_test.map	2014/07/01 11:06	MAP ファイル	236 KB	
1 コンピューター	-	soft_test.objdump	2014/07/01 11:06	OBJDUMP ファ	611 KB	
mem_init ファイル フォル	更新 ダー	行日時: 2014/07/01 11:05				



mem\_init フォルダ内には、下図のように、meminit.qip ファイルと nios2\_system\_onchip\_memory.hex ファイル (※このファイル名は、プロジェクトごとに異なります。)を確認することができます。

					X
	_14_0 • software • soft_test • mem_init	► <b>-</b> 4 <sub>7</sub>	mem_initの検索		٩
整理 ▼ 🐻 開く ▼ 書き	込む 新しいフォルダー				?
📃 デスクトップ 🔷	名前	更新日時	種類	サイズ	
📃 最近表示した場所	🚯 hdl_sim	2014/07/01 11:06	ファイル フォル…		
	📓 meminit.qip	2014/07/01 11:05	QIP ファイル	1 KB	
🍃 ライブラリ	] meminit.spd	2014/07/01 11:05	SPD ファイル	1 KB	
📑 ドキュメント	nios2_system_onchip_memory.hex	2014/07/01 11:06	HEX ファイル	309 KB	
■ ピクチャ					
🛃 ビデオ					
⊿) ミュージック					
-					
nios2_system_onchi HEX ファイル	p_memory.hex 更新日時: 2014/07/01 11:06 サイズ: 308 KB	作成日時: 2014/0	7/01 10:59		

nios2\_system\_onchip\_memory.hex ファイルをテキスト・エディタで開くと、Hex フォーマットでの記述となりますが、 ソフトウェア・コードがデータとして反映されていることが確認できます。





#### 2-3. Quartus II での設定とコンパイル

Assignments メニューから Settings を選択します。

🍇 Quartus II 64-Bit - C:/	Lab/	nios2_basic_prj_14_0/nios2	_basic_lab - ni	ios2_basic_l
File Edit View Project	Assi	gnments Processing Tools \	Window Help	Ţ
i 🗋 💕 💭 🍠 🕺 🖬	Ŷ	Device		1 🕅 😢 .
Project Navigator	2	Settings	Ctrl+Shift+E	Compilation
	4	Assignment Editor	Ctrl+Shift+A	e of Contents
A Cyclone V: 5CGXFC5C6F27	<b>&gt;</b>	Pin Planner	Ctrl+Shift+N	Flow Sum
🔺 🔛 nios2_basic_lab 🛅		Remove Assignments		📰 Flow Sett
방법_sld_hub:auto_hub	12	Back-Annotate Assignments		Flow Non
▷ abd nios2_system:inst	- /	Import Assignments		Flow Elap
				Flow OS 9

下図のように、Settings 画面が起動します。画面左側の Category 欄から Files を選択して、右側の File name 欄に、前述の meminit.qip ファイルを指定した後、[Add] ボタンをクリックして追加登録を行います。

下図のように、File name 枠内に追加されていることが確認できたら、[OK] ボタンをクリックします。

tegory:				Device
General Files Libraries IP Settings IP Catalog Search Locations Operations and Conditions	Files Select the design files you want to include in project. File name:	he project. Click Add All to add all desi	gn files in the project di	rectory to the
Voltage     Temperature     Compilation Process Settings     Incremental Compilation     Physical Synthesis Optimizations     EDA Tool Settings     Design Entry/Synthesis     Simulation     Formal Verification     Board-Level     Analysis & Synthesis Settings     VHDL Input     Verilog HDL Input     Default Parameters     Elter Settings	File Name <u>software/soft_test/mem_init/meminit.gip</u> my_onginai_constrain.sdc ▷ nios2_system/synthesis/nios2_system.g nios2_basic_lab.bdf	Type L IP Variation File (.qip) Synopsys Design Constraint ip IP Variation File (.qip) Block Diagram/Schematic File	ibrary Design Entry, <none> <none> <none> <none></none></none></none></none>	Add Al



Processing メニューから Start Compilation を選択して、フル・コンパイルを行います。



コンパイルが完了すると作成した HEX ファイルの内容を含んだ SOF/POF ファイルが生成されるので Quartus II Programmer で SOF ファイルを書き込み、実機上で正常に動作するか確認してください。本プログラ ムでは、SOF を書き込んだ時点で Nios II が制御する LED が規定の動作を行っていることを確認できました。 正常に動作が確認できれば、FPGA 内のオンチップメモリに Nios II のプログラムが正常に初期化されていること になります。

#### 2-4. HEX ファイル更新時の Quartus II プロジェクトへの反映方法

Nios II の実行ファイルである ELF ファイルが変更された際には、2-1 章の手順により HEX ファイルを更新し ますが、そのたびに Quartus II をフル・コンパイルするのは効率的ではありません。ここでは、HEX ファイルに対 応するオンチップ・メモリの内容のみ更新する方法をご紹介します。

まず、Processing メニューから Update Memory Initialization File を選択します。この処理で更新された HEX ファイルを解析してファイルに問題があるかどうかをチェックします。この時点では、HEX ファイルの内容は SOF/POF ファイルには反映されないことに注意してください。



正常に解析されると、Quartus II のメッセージが以下のように表示されます。

AI 😢	🛆 🔺	Control Con
Туре	ID	Message
1		***************************************
Þ 🛈 👘		Running Quartus II 64-Bit MIF/HEX Update
<u>(</u> )		Command: quartus_cdb nios2_basic_lab -c nios2_basic_labupdate_mif
Á	39023	Can't find Memory Initialization File nios2_system_nios2_opu_ic_tag_ram.mif skipped updates for this file
<u> </u>	39023	Can't find Memory Initialization File nios2_system_nios2_opu_bht_ram.mif skipped updates for this file
<u>A</u>	39023	Can't find Memory Initialization File nios2_system_nios2_opu_rf_ram_a.mif skipped updates for this file
<u>A</u>	39023	Can't find Memory Initialization File nios2_system_nios2_opu_rf_ram_b.mif skipped updates for this file
<u>A</u>	39023	Can't find Memory Initialization File nios2_system_nios2_cpu_dc_tag_ram.mif skipped updates for this file
Δ	39023	Can't find Memory Initialization File nios2_system_nios2_cpu_ociram_default_contents.mif skipped updates for this file
<u>A</u>	39023	Can't find Memory Initialization File nios2_system_epcq_boot_rom.hex skipped updates for this file
4 🕕 👘	39024	Processed the following Memory Initialization File(s)
- i 🕡	39025	Processed Memory Initialization File C:/Lab/nios2 basic prj 13 1/software/soft test/mem init/nios2 system onchip memory.hex
Þ 🕕 🗌		Quartus II 64-Bit MIF/HEX Update was successful. 0 errors, 7 warnings

次に、Processing メニューから Start ⇒ Start Assembler を選択します。この処理で SOF/POF ファイルに新し い HEX ファイルの内容が反映されます。フル・コンパイルに比べて、FPGA 内部のオンチップ・メモリのみ更新 するので処理が短時間で済みます。

😻 Quartus II 64-Bit - C:/Lab/nios2_basic_prj_14_0/nios2_basic_lab - nios2_basic_lab								
File Edit View Project Assignments	Proc	essing Tools Window Help 🗟						
9 (* 🖬 🖬 👗 🖬 🛍 🗐 (*	5707	Stop Processing	Ctrl+Shift+C	1	) 💿 🕨 🤣 😓 🕲 💘	😤 \ominus 🐌 🚣 🧕	Ţ	
Project Navigator		Start Compilation	Ctrl+L	os2_	basic_lab 🔀 🛛 📸 nios2_ba	asic_lab.bdf 🛛 🔀		
	2	Analyze Current File			I 🗗 Flow Summary			
Cyclone V: 5CGXFC5C6F27C7		Start	+	$\triangleright_{\rm A}$	Start Hierarchy Elaboration		Tu	
▲ 🔐 nios2_basic_lab 🔤		Update Memory Initialization File		5	Start Analysis & Elaboration		20	
abd_sid_hub:auto_hub	٩	Compilation Report	Ctrl+R	i.	Start Analysis & Synthesis	Ctrl+K	la	
▷ abd nios2_system:inst	٩	Dynamic Synthesis Report		14	Start Partition Merge		E2	
	4	PowerPlay Power Analyzer Tool		ыü	Start Fitter			
<	W	SSN Analyzer Tool	L	N.	Start Assembler		080	
A Hierarchy Files Design U		Receive Compilation Status Notifications		8 12	Start TimeQuest Timing Analyzer Start EDA Netlist Writer	Ctrl+Shift+T	3 9	



以下のメッセージが表示される場合があるので、Yes をクリックし処理を進めます。



この処理が完了したら、SOF/POF ファイルのタイム・スタンプが更新されていることを確認してください。また、改めて FPGA に書き込んで実機での動作を確認してください。

## <u> 改版履歴</u>

Revision	年月	概要
1	2014 年 7 月	新規作成
2	2014 年 9 月	ブート用オンチップ・メモリが ROM の場合を追加

#### 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。