

Nios II Processor Booting from CFI Flash on Cyclone V GX FPGA Development Kit

Date: 29 March 2017

Revision: 1.0

©2017 Intel Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, INTEL, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Intel Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Intel warrants performance of its semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel 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 Intel. Intel 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.

Introduction

This document demonstrates step-by-step instructions on how to boot a simple Hello World Nios II application from CFI flash using <u>Cyclone V GX FPGA Development Kit</u>.

Prerequisite

Before you start, download the <u>Kit installation</u> for Cyclone V GX FPGA Development Kit.

Related links:

- <u>Avalon Tri-State Conduit Components User Guide</u>
- Parallel Flash Loader IP Core User Guide
- <u>Generic Nios II Booting Methods User Guide</u>

Design Creation

In this example, you do not have to create your hardware design from scratch, use the hardware design in "<kit installation directory>/cycloneVGX_5cgxfc7df31es_fpga/examples/board_update_portal". Ensure Generic Tri-State Controller and Tri-State Conduit Bridge are added in the Qsys system (c5gxfc7_fpga_bup_qsys.qsys). You may remove other unnecessary components such as TSE to simplify the design.

\subset					Avaion memory mapped master	υσουρια-επιεκ το export	[CIK]	
				🗆 сри	Nios II (Classic) Processor			
				→ dk	Clock Input	Double-click to export	clkin_50	
		+ +	+	→ reset_n	Reset Input	Double-click to export	[clk]	
	\succ			≺ data_master	Avalon Memory Mapped Master	Double-click to export	[clk]	
	\vdash		_	instruction_master	Avalon Memory Mapped Master	Double-click to export	[dk]	
			_	→ d_irq	Interrupt Receiver	Double-click to export	[dk]	
			_	<pre>itag_debug_module_r</pre>	Reset Output	Double-click to export	[dk]	
Þ	ا ♦ ا		_	→ jtag_debug_module	Avalon Memory Mapped Slave	Double-click to export	[dk]	
		×		 custom_instruction_m 	. Custom Instruction Master	Double-click to export		
				onchip_ram	On-Chip Memory (RAM or ROM)			
				→ dk1	Clock Input	Double-click to export	clkin_50	
	ا ♦ ا		_	→ s1	Avalon Memory Mapped Slave	Double-click to export	[dk1]	
		• • • •	- +	→ reset1	Reset Input	Double-click to export	[dk1]	
	ا ♦ ا		_	→ s2	Avalon Memory Mapped Slave	Double-click to export	[dk2]	
				→ dk2	Clock Input	Double-click to export	clkin_50	
		¢	- +	→ reset2	Reset Input	Double-click to export	[dk2]	
				🗆 tristate_conduit_bri.	Tri-State Conduit Bridge			
				→ dk	Clock Input	Double-click to export	clkin_50	
		¢	-	→ reset	Reset Input	Double-click to export	[dk]	
		• • •		→ tcs	Tristate Conduit Slave	Double-click to export	[dk]	
				≻ out	Conduit	tristate_conduit_bridge		
				🗆 🛄 tristate_conduit	Tri-State Conduit Pin Sharer			
			++	→ dk	Clock Input	Double-click to export	clkin_50	
		↓ ↓ ↓	+	→ reset	Reset Input	Double-click to export	[dk]	
				≺ tcm	Tristate Conduit Master	Double-click to export	[clk]	
			_	→ tcs0	Tristate Conduit Slave	Double-click to export	[dk]	
		↓		→ tcs1	Tristate Conduit Slave	Double-click to export	[clk]	
				□ 喧 ext_flash	Generic Tri-State Controller			
			++	→ dk	Clock Input	Double-click to export	clkin_50	
		↓ ↓	+	→ reset	Reset Input	Double-click to export	[clk]	
Þ	ا ♦ ا			→ uas	Avalon Memory Mapped Slave	Double-click to export	[dk]	≜ 0x0000_0000
				≺ tcm	Tristate Conduit Master	Double-click to export	[dk]	
				🗆 🛄 ssram	Generic Tri-State Controller			
				→ dk	Clock Input	Double-click to export	clkin_50	
		¢ •	+	→ reset	Reset Input	Double-click to export	[clk]	
	• •			→ uas	Avalon Memory Mapped Slave	Double-click to export	[clk]	■ 0x0700_0000
		4		≺ tcm	Tristate Conduit Master	Double-click to export	[clk]	
				merged_resets	Reset Bridge			

Qsys Setting

As we are using the Board Update Portal example, the factory hardware and software block in CFI flash will be replaced with new hardware and software blocks created here.

In Nios II Processor parameter editor, set the reset vector memory and exception vector memory as below:

leset Vector		
eset vector memory:	ext_flash.uas	\sim
eset vector offset:	0x01bc0000	
leset vector:	0x01bc0000	
Exception Vector		
exception vector memory:	onchip_ram.s1	\sim
Exception vector offset:	0x0000020	
Exception vector:	0×08200020	

Quartus Setting

The MSEL pins in Cyclone V GX development kit are defaulted to FPPx16 configuration scheme. In Quartus Prime software, set the configuration settings as below:

Configuration										
Specify the device configuration scheme and the configuration device.										
Configuration scheme: Passive Parallel x16										
Configuration mode:	Standard		•							
Configuration device										
Use configuration	device:	Auto 👻								

BSP Editor Setting

In Nios II SBT tool, create the Nios II processor HAL BSP based on the .sopcinfo created from Qsys generation. Refer to table below for BSP editor settings:

Boot Option	BSP Editor Setting: Linker Script	BSP Editor Seting: Settings.Advanced.hal.linker
Nios II processor application execute- in-place from CFI flash	 Set .text Linker Section to CFI flash Set other Linker Sections (.heap, .rwdata, .rodata, .bss, .stack) to OCRAM/ External RAM 	If the exception vector memory is set to OCRAM/ External RAM, enable the following settings: allow_code_at_reset enable_alt_load enable_alt_load_copy_rodata enable_alt_load_copy_rwdata enable_alt_load_copy_exceptions If the exception vector memory is set to CFI flash, enable the following settings: allow_code_at_reset enable_alt_load enable_alt_load enable_alt_load enable_alt_load
Nios II processor application copied from CFI flash to RAM using boot copier	All Linker Sections are set to OCRAM/ External RAM	All settings are left unchecked

Application

- 1. In the Nios II SBT, create a new application using Hello World template.
- 2. Once the BSP editor settings configured, build the project and ELF file will be created.
- 3. Use mem_init_generate utility to generate FLASH file by selecting Make Targets \rightarrow Build.
- 4. Launch Nios II Command Shell, use nios2-elf-objcopy utility to generate HEX file.

opy from `ext_flash.flash' [srec] to `ext_flash.hex' [ihex]

Programming Files Generation

This example demonstration is using Board Update Portal design for the development kit, therefore you need to follow the CFI flash memory map as below:

	Block Description	KB Size	Address Range
	Unused	128	0x03FE.0000 - 03FF.FFFF
	User software	28,800	0x023C.0000 - 03FD.FFFF
	Factory software	8192	0x01BC.0000 - 023B.FFFF
•	zipfs (html, web content)	4096	0x017C.0000 - 01BB.FFFF
	User hardware 2	8064	0x00FE.0000 - 017B.FFFF
	User hardware 1	8064	0x0080.0000 - 00FD.FFFF
	Factory hardware	8064	0x0002.0000 - 007F.FFFF
	PFL option bits	32	0x0001.8000 - 0001.FFFF
	Board information	32	0x0001.0000 - 0001.7FFF
	Ethernet option bits	32	0x0000.8000 - 000.FFFF
	User design reset vector	32	0x0000.0000 - 000.7FFF

To generate the POF file, follow the configuration as below:

🖹 Convert Programming File - C:/intelFPGA/16.1/local/nios_booting/cfi/cvgx_cfi/c5gxfc7_fpga_bup - c5gxfc7_fpga_bup - 🛛 🗙									
<u>F</u> ile <u>T</u> ools <u>W</u> indow						Search altera	a.com		
Specify the input files to o You can also import inpu future use. Conversion setup files	convert and the type t file information fro	e of programming file to ge om other files and save the	nerate. conversion setup	information cre	ated here for				
Ор	Open Conversion Setup Data Save Conversion Setup								
Output programming fi	le								
Programming file type:	Programmer Obje	ct File (.pof)					•		
Options/Boot info	Configuration devi	ce: CFI_512Mb	•	<u>M</u> ode:	Passive	Parallel x16	-		
File <u>n</u> ame:	output_file.pof								
Advanced	Remote/ <u>L</u> ocal upd	ate difference file: NO	ONE				~		
	Create Memory	Map File (Generate output	_file.map)	file and a file					
	Create CvP files	; (Generate output_file.perip	on.pot and output	_file.core.rbf)					
Input files to convert		ala ni b (achelate output_)							
File/Data :	area	Properties	Start Addres	s		A	dd He <u>x</u> Data		
Options			0x00018000			A	dd Sof Page		
✓ Hex Data	A	bsolute addressing	0x01BC0000				Add File		
ext_flash.hex ✓ SOF Data	Pa	age O	0x00020000				Demonstra		
c5gxfc7_fpga_	bup.sof 50	GXFC7D6F31ES					Remove		
							Up		
							Down		
							Propert <u>i</u> es		
					<u>G</u> enerate	Close	Help		

Programming CFI Flash

PFL IP core is required to program the CFI flash. A simple PFL design output file is available here.

Programmer - C:/intelFPGA/16.1/local/nios_booting/cfi/cvgx_cfi/c5gxfc7_fpga_bup - c5gxfc7_fpga_bup - [c5gxfc7_fpga_bup.cdf]* _ \times <u>File Edit View Processing Tools Window Help</u> 6 Search altera.com Aardware Setup... USB-BlasterII [USB-1] Mode: JTAG • Progress: 100% (Successful) $\hfill\square$ Enable real-time ISP to allow background programming when available Device Security File Checksum Usercode Program/ Verify Blank-Examine Erase ISP IPS File Mart Start Configure Check Bit CLAMP Stop <none> 5CGXFC7D6 00000000 <none> misc/maxV_pfl.pof 5M2210ZF256 009DADF1 009DADD9 \checkmark 💏 Auto Detect \checkmark CFM 🗙 Delete UFM \checkmark 💾 Add File... 隆 Change File... Save File 📌 Add Device... ------TD 1¹ Up ↓[™] Down 5CGXFC7D6 5M2210ZF256 TDO

1. Program the "maxV_pfl.pof" into the MAX V device.

- 2. Click Auto Detect and you should see CFI_512Mb.
- 3. Program the "output_file.pof" into the CFI flash.

Programmer - C:/intelFPGA/16.1/local/nios_booting/cfi/csgxfc7_fpga_bup - c5gxfc7_fpga_bup - [c5gxfc7_fpga_bup.cdf]* — — X Eile Edit View Processing Lools Window Help Search altera.com •													
Ardware Setup USB-Blasteril (USB-1)						Mode: JTAG				ogress: (100% (Successful)		
Enable real-time ISP to allow background programming when available													
File Device Checksum Usercode				Program/ Configure	Verify	Blank- Check	Examine	Security Bit	Erase	ISP CLAMP	IPS File		
■ ¹ ¹ ¹ Stop	<none></none>	5CGXFC7D6	00000000	<none></none>									· · ·
Auto Detect	<none></none>	5M2210ZF256	00000000	00000000									
	✓ output_file.pof	CFI_512Mb	9D7A64D1		\checkmark								
X Delete	Page_0												
🏓 Add File	ext_flash.hex												
Change File	OPTION_BITS				\checkmark								
Save File Add Device Î [™] Up I [™] Down	TDI 5CGXFC7	GFI_511 → 5M22102	ZF256										

4. Restore the MAX V design by programming the "**max5.pof**" in "*<kit installation directory>/ cycloneVGX_5cgxfc7df31es_fpga/factory_recovery*".