

# Nios II 簡易チュートリアル

ver.14



2014年8月 Rev.1

ELSENA,Inc.





# Nios II 簡易チュートリアル

# <u>目次</u>

1. はじめに	3
1-1. フロー概要	
2. ハードウェア	4
2-1. ハードウェア・プロジェクトの作成 (Quartus II)	4
2-2. コンフィギュレーション・モードの設定	5
2-3. Qsys にてシステムを構成し、HDL を生成	6
2-4. Quartus II にてコンパイル	11
2-5. システムを FPGA ヘダウンロード	14
3. ソフトウェア	15
3-1. Nios II SBT でソフトウェア・プロジェクトの作成	15
3-2. 実行コードをターゲットで実行、デバッグ	19
改版履歴	22

#### 1. <u>はじめに</u>

この資料は、Quartus<sup>®</sup> II 14.0、Nios<sup>®</sup> II 14.0 Software Build Tools for Eclipse (以降、SBTと略)を使用して、 Nios II のシステム生成をチュートリアル形式でまとめたものです。シンプルなハードウェア、ソフトウェアを使用し、 動作確認を行うまでの最低限必要となる開発フローをご理解いただけます。ハードウェアの構成は、Nios II コア、 JTAG UART、オンチップメモリ、LED 用の8ビット出力用の PIO コンポーネントのみです。ソフトウェアは、LED を点滅させるだけの、非常にシンプルな記述です。チュートリアルのフローを以下に示します。

#### 1-1. フロー概要

- ①Quartus II の起動
- ② Qsys にて、Nios II システムの構築、生成
  ③ Quartus II にてハードウェアデザインのコンパイル
  ④ ターゲット・ボードへ .sof ファイルをダウンロード
  ⑤ Nios II SBT にて、ソフトウェア・プロジェクトの作成
  ⑥ ソフトウェア・ソースファイルを新規で作成し、記述
  ⑦ ソフトウェア・ソースファイルのビルド
  ⑧ 生成されたソフトウェアの実行ファイルをターゲット上で実行
  ⑨ デバッガを起動し、デバッグ

この資料では、Terasic 社の Cyclone V GX スターター開発キットを使用します。他の評価ボードを使用する場合は、Quartus II にて設定するデバイス型名の選択、ピン配置、未使用ピンの処理等をそれぞれのボードの仕様に合わせて行う必要があります。



#### 2. <u>ハードウェア</u>

- 2-1. ハードウェア・プロジェクトの作成 (Quartus II)
  - ① Quartus II を起動し、Quartus II の File メニュー  $\rightarrow$  New Project Wizard をクリックします。

New Project Wizard の Introduction ウインドウが表示されたら、Next を 1 回クリックし、以下の画面で、 ワーキング・ディレクトリ、ハードウェア・デザインのプロジェクト名、ハードウェア・デザインのトップのエン ティティ名を入力します。

💱 New Project Wizard	×
Directory, Name, Top-Level Entity [page 1 of 5]	
What is the working directory for this project?	
C:\Lablnios2 basic pri	
What is the name of this project?	
nios2_basic_lab	
What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.	
nios2_basic_lab	
Use Existing Project Settings	
< <u>B</u> ack <u>N</u> ext > Einish Cancel <u>H</u> e	ip 🛛

② Next を 2 回クリックし、以下の画面で、ターゲット・デバイスの選択をします。

ここでは、C yclone V を選択し、Available devices 中の 5CGXFC5C6F27C7 を選択します。

Family & De Select the family and rou can install addit To determine the ve Device family Eamily: Cyclone Devices: All	d device you want to ional device support v rsion of the Quartus V (E/GX/GT/SX/SE/ST	s [page target for co with the Insta II software in	3 of 5] mpilation. Il Devices comman n which your targe	d on the Tools menu. t device is supported, refer Show in 'Available Package: Pin gount:	to the <u>Device Support List</u> devices' list <u>Any</u> <u>Any</u>	webpage.
Auto device si     Auto device si     Specific device     Qther: n/a	elected by the Fitter a selected in 'Available	e devices' list		Name filter:	scgxFC5C6F27C7 ed devices	
-	Core Voltage	ALMs	User I/Os	GXB Channel PMA	GXB Channel PCS	PCIe (PIPE) Hard IP Bloc
Name			0.54			
Name 5CGXFC5C6F27C7	1.1V	29080	364	6	6	2
Name 5CGXFC5C6F27C7	1.1V	29080	364	6	6	2

- Next を 2 回クリックし、表示内容が正しいことを確認して Finish でこのウインドウを閉じます。
   この段階で、ハードウェア・プロジェクトが作成されます。
- 2-2. コンフィギュレーション・モードの設定
  - ① Assignments  $\forall$ = $\neg$ → Device  $e^{j}$ , Device and Pin Options  $e^{j}$ ,  $e^{j}$ , e

o determine the ver	sion of the Qualitus II	sortware in v	nici your target	device is supported, i		ule <u>Device Support List</u> w	eupage.
Device family				Show in 'Available de	evices' li	st	
Eamily: Cyclone V (E/GX/GT/SX/SE/ST)			-	Package:	Any		•
Devices: All				Pin count:	Anv		•
Devices: Aii			•		,		
Target device				Core Speed grade:	Any		•
- a get de lice				Name filter:			
Auto device sel	ected by the Fitter			Chow advanced	douicco		
Specific device:	selected in 'Available o	levices' list		Stiow advanced	uevices		
Other: n/a			<b>_</b>	a i la a r			
				Device and Pin Option	ns		
vailable devices:							
Name	Core Voltage	ALMs	User I/Os	GXB Channel I	РМА	GXB Channel PCS	PCIe (I
5CGXFC4C7M13C8	1.1V	18860	203	6		6	2
5CGXFC4C7U19C8	1.1V	18860	252	6		6	2
5CGXFC4F6M11C6	1.1V	18860	149	4		4	1
5CGXFC4F6M11C7	1.1V	18860	149	4		4	1
5CGXFC4F6M11I7	1.1V	18860	149	4		4	1
5CGXFC4F7M11C8	1.1V	18860	149	4		4	1
5CGXFC5C6F23A7	1.1V	29080	268	6		6	2
5CGXFC5C6F23C6	1.1V	29080	268	6		6	2
5CGXFC5C6F23C7	1.1V	29080	268	6		6	2
5CGXFC5C6F23I7	1.1V	29080	268	6		6	2
5CGXFC5C6F27C6	1.1V	29080	364	6		6	2
FOOVEOFOOD	1.1V	29080	364	6		6	2
SCGXFCSC6F2/C/	1.1V	29080	364	6		6	2
5CGXFC5C6F27L7	1.1V	29080	203	6		6	2
5CGXFC5C6F27L7 5CGXFC5C6F27L7 5CGXFC5C6M13C6	1 1V	29080	203	6		6	2
5CGXFC5C6F27L7 5CGXFC5C6F27L7 5CGXFC5C6M13C6 5CGXFC5C6M13C7	1.1.	29080	203	6		6	2
5CGXFC5C6F27C7 5CGXFC5C6F27I7 5CGXFC5C6M13C6 5CGXFC5C6M13C7 5CGXFC5C6M13I7	1.1V		252	6		6	2 .
5CGXFC5C6F27I7 5CGXFC5C6F27I7 5CGXFC5C6M13C6 5CGXFC5C6M13C7 5CGXFC5C6M13I7 5CGXFC5C6U19A7	1.1V 1.1V 1.1V	29080	252				

 Device and Pin Options ウインドウの Configuration を開き、Configuration scheme で Active Serial x1 を 選択し、OK をクリックします。

S Device and Pin Options - nios2_basic_	_lab 📃	
Category:		
Category: General Configuration Programming Hiles Unused Pins Dual-Purpose Pins Capacitive Loading Board Trace Model I/O Timing Voltage Pin Placement Error Detection CRC CVP Settings Partial Reconfiguration	Jab         Configuration         Specify the device configuration scheme and the configuration device.         Configuration gcheme         Active Serial x1 (can use Configuration Device)         Configuration mode:         Standard         Configuration device:         Image: Configuration device:         Image: Quest configurat	
	Description: Specifies the configuration mode used with the configuration scheme for configuring the device. Reset OK Cancel Help	

- 2-3. Qsys にてシステムを構成し、HDL を生成
  - ① Quartus II の Tools メニュー  $\rightarrow$  Qsys をクリックし、Qsys を開きます。

File ⇒ Save as... として先にファイルにセーブします。今回は、nios2\_system.qsys というファイル名で保存します。

🙏 Qsys	a Hala					
	S getp	ess Map 🙁 Interconnect Requi	rements 🙁 Device Family 🌣	3		
Project 	Use Corn Name Use Corn Name CR	Description Clock Source Olock Input Reset Input Clock Output Reset Output	Export Clk reset Dauble-click to	Clock export o export ck.0	Bace	
New Edt	d ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓	▲ 保存 (保存: ● nic ● db 最近使った項 ■ デスクトップ マイドキュメント ● コンピューター	s2,basic_prj		• 3 1	
S		です。 ネットワーク ファイルタ	5: Inios2_system.qsys ログ: Qsys System Files (	(*.gsys)	•	保存(S) 取消し

 Qsys 左枠の IP Catalog より Basic Functions グループ内の On-Chip Memory カテゴリにて On-Chip Memory (RAM or ROM) をダブルクリックします。

On-Chip Memory (RAM or ROM) ウインドウ内の Total memory size を 128 K Bytes ("128k"と入力で 反映されます) に設定し Finish をクリックします。

👃 On-Chip Memory (RAM or ROM	1) - onchip_memory2_0		×
On-Chip Memory altera_avalon_onchip_mem	(RAM or ROM) ory2		Documentation
Block Diagram     Show signals     onchip_memory2_0     cik1    olock     g1    avaion	Memory type Type:     Dual-port access     Single clock operation Read During Write Mode: Block type:	RAM (Writeble) • DONT_CAR_ • AUT_ •	
atera_avalon_onchip_memory2	▼ Size Data width: Total memorysize: Mnimize memoryblock usage (may in	Iso bytes pact (max)	
	Read latency     Slave s1 Latency     Slave s2 Latency     Very State s2 Latency     Court Add Memory Protection     Death Operation		
	ECC Parameter     Extend the data width to support ECC bits	Enabled •	
	Memory initialization     Initialize memory content     Enable non-default initialization file     Type the filename (e.g: my_ramhex	) or select the hex file using the file browser button.	
	User created initialization file: Enable In-System Memory Content Ed Instance ID:	onchip_mem.hex for feature NONE	
	Memory will be initialized from nice	;2_system_onchip_memory2_0.hex	
		Car	cel Finish

③ Qsys 左枠の IP Catalog より、Processors and Peripherals グループ内の Embedded Processors カテゴリに て Nios II Processor をダブルクリックし Nios II Processor のウィザード内で Nios II コア を選択し Finish をクリックします。

how signals	Core nos a Caches and Memory Int		1000 (1000) (C		
		terfaces   Advanced Features	MMU and MPU Settings   JTA	G Debug Module	
	Select a Nios II Core				
hios2_dsys_0	1000 1 0000	Nios II/e			
clock avaion data_master_					
et_n instruction_master		NIOS I/1			
rg #ag_debug_module_reset	N	√ios II∕e	Nios II/s	Nios II/f	1
_debug_module custom_instruction_master_	Nine II B	usc	RISC	RISC	-
avaion nos_cosom_norcoor	NIOS II 3	2-bit	32-bit	32-bit Instruction Cashe	
atera_wosz_dsys			Branch Prediction	Branch Prediction	
			Hardware Multiply Hardware Divide	Hardware Multiply Hardware Divide	
				Barrel Shifter	
				Dynamic Branch Prediction	
	Memory Usage (e.g. Stratix IV) To	wo M9Ks (or equiv.)	Two M9Ks + cache	Three M9Ks + cache	
	Hardware Arithmetic Operation	Embedded Multiplie	-1		
	Hardware divide	"Employed Humphie"			
	* Reset Vector				
	Reset vector memory:	None	•		
	Reset vector offset	0x00000000			
	Reset vector:	0×00000000			
	* Exception Vector				
	Exception vector memory	None	•		
	Cheepinal rectar memory				
	Exception vector offset	0x00000020			
	Exception vector include p Exception vector:	0x000000000000000000000000000000000000			
	Exception vector offset Exception vector:	0x00000020 0x00000000			

 ④ Qsys 左枠の IP Catalog より、Processors and Peripherals グループ内の Peripherals カテゴリより PIO(Parallel I/O) をダブルクリックします。こちらのウィザードはデフォルトの設定で Finish をクリックしま

width (1-32 bits):       8         pio_0       Direction:         set       Isset         it       avaian         it       avaian         internal_connection       ondult         aters_avaian_pio       Enable individual bit settine/clearing          Edge appure register          Synchronously capture         Enable bit-clearing for edge capture register         Interrupt         Generate IR0         IR0 Type:       LEV         Level: Interrupt CPU when any unmasked I/O pin is logic true         Edge: interrupt CPU when any unmasked I/O pin is logic true         Edge: interrupt CPU when any unmasked I/O pin is logic true         Edge: interrupt CPU when any unmasked I/O pin is logic true         Edge: interrupt CPU when any unmasked I/O pin is logic true         Edge: interrupt CPU when any unmasked I/O pin is logic true         Edge: interrupt CPU when any unmasked I/O pin is logic true         Edge: interrupt CPU when any unmasked I/O pin is logic true         Edge: interrupt CPU when any unmasked I/O pin is logic true         Edge: interrupt CPU when any unmasked I/O pin is logic true         Edge: interrupt CPU when any unmasked I/O pin is logic true         Edge: interrupt CPU when any unmasked I/O pin is logic true	ck Diagram	Basic Settings
pio_0         :k         eset         ist         ist <td< td=""><td>w signals</td><td>Width (1-32 bits): 8</td></td<>	w signals	Width (1-32 bits): 8
ak       obok         eset       reset         avaion       output Port Reset Value: 0x000000000000         output Port Reset Value: 0x0000000000000         • Dutput Register         • Enable individual bit setting/clearing         • Edge capture register         • Synchronously capture         Edge Type:         • Enable bit+clearing for edge capture register         • Interrupt         • Generate IRO         IRO Type:       LEV         Levet Interrupt CPU when any unmasked J/O pin is logic true         • Edge: interrupt CPU when any unmasked J/O pin is logic true         • Edge: interrupt CPU when any unmasked J/O pin is logic true         • Edge: interrupt CPU when any unmasked J/O pin is logic true         • Hardwire PIO inputs in test bench.         • Hardwire PIO inputs in test bench.	pio 0	Direction: O Bidir
attera_avalon_plo         attera_avalon_plo <t< td=""><td></td><td>🔘 Input</td></t<>		🔘 Input
eset  A avaion  Atternal_connection  eonduit  atterna_avaion_pio	clock	InOut
avaien       avaien         conduit       atters_avaien_pio         atters_avaien_pio       Output Port Reset Value: 0x00000000000000000000000000000000000	set reset	Output
wternal_connection       conduit         attern_avalen_pio         attern_avalen_pio         Chaptit Register         Chaptit Register <td< td=""><td>avalon</td><td>Output Port Reset Value: 0x000000000000000</td></td<>	avalon	Output Port Reset Value: 0x000000000000000
atera_avalen_pio          Chable individual bit setting/clearing             Edge capture register           Synchronously capture             Edge Type:           RISING -             Chable bit-clearing for edge capture register               Chable bit-clearing for edge capture register               Chable bit-clearing for edge capture register               Charrupt           Cherrupt OPU when any unmasked I/O pin is logic true             Edge: Interrupt OPU when any unmasked bit in the edge-capture           Level: Interrupt OPU when any unmasked bit in the edge-capture             Level: Interrupt OPU when any unmasked I/O pin is logic true           Edge: Interrupt OPU when any unmasked bit in the edge-capture             Level: Interrupt OPU when any unmasked I/O pin is logic true           Edge: Cherrupt OPU when any unmasked I/O pin is end             Level: Interrupt OPU when any unmasked I/O pin is logic true           Edge: Cherrupt OPU when any unmasked I/O pin is end             Level: Interrupt OPU when any unmasked I/O pin is interventer           Edge: Cherrupt OPU when any unmasked I/O pin is intervente	ternal_connection conduit	Output Register
	altera_avalon_pio	Enable individual bit setting/clearing
Synchronously capture Edge Type: RISING  Enable bit-clearing for edge capture register  Interrupt Generate IRQ IRQ Type: Level: Interrupt CPU when any unmasked J/O pin is logic true Edge: Interrupt CPU when any unmasked bit in the edge-capture register is logic true. Available when synchronous capture is enable Test bench wring Hardwire PIO inputs in test bench. Disis inputs to:		TEdge capture register
Edge Type:       RISING         Enable bit-clearing for edge capture register         Interrupt         Generate IRQ         IRQ Type:       LEV         Levet: Interrupt CPU when any unmasked J/O pin is logic true         Edge: Interrupt CPU when any unmasked bit in the edge-capture register is logic true. Available when synchronous capture is enable         ** Test bench wiring         Hardwire PIO inputs in test bench.         Drive inputs true		Synchronously capture
Enable bit-clearing for edge capture register      Interrupt      Generate IRQ      IRQ Type: LEV Levet: Interrupt CPU when any unmasked J/O pin is logic true     Edge: Interrupt CPU when any unmasked bit in the edge-capture     resister is logic true. Available when synchronous capture is enable      Test bench wiring     Hardwire PIO inputs in test bench.      Drive inputs to test		Edge Type: RISING -
Interrupt  Generate IRQ  IRQ Type: Level: Interrupt CPU when any unmasked J/O pin is logic true Edge: Interrupt CPU when any unmasked bit in the edge-capture resister is logic true. Available when synchronous capture is end  Test bench wiring Hardwire PIO inputs in test bench. Drive inputs to test bench.		Enable bit-clearing for edge capture register
Generate IRQ     IRQ Type: LEV Levet: Interrupt CPU when any unmasked I/O pin is logic true     Edge: Interrupt CPU when any unmasked bit in the edge-capture     resister is logic true. Available when synchronous capture is enab     Test bench wiring     Hardwire PIO inputs in test bench.     Drive inputs to the test bench.		* Interrupt
IRQ Type: LEV Level: Interrupt CPU when any unmasked I/O pin is logic true Edge: Interrupt CPU when any unmasked bit in the edge-capture register is logic true. Available when synchronous capture is enab  Test bench wiring Hardwire PIO inputs in test bench. Disin inputs to test bench.		🗌 Generate IRQ
Level: Interrupt CPU when any unmasked I/O pin is logic true Edge: Interrupt CPU when any unmasked bit in the edge-capture register is logic true. Available when synchronous capture is enab		IRQ Type:
Test bench wiring     Hardwire PIO inputs in test bench     Drive invite tv		Level: Interrupt CPU when any unmasked I/O pin is logic true Edge: Interrupt CPU when any unmasked bit in the edge-capture register is logic true. Available when synchronous capture is enabled
Hardwire PIO inputs in test bench		* Test bench wiring
Drive inputs to:		Hardwire PIO inputs in test bench
Unive inputs to: 0x0000000000000		Drive inputs to: 0x00000000000000000000000000000000000

す。

⑤ Qsys 左枠の IP Catalog より、Interface Protocols グループ内の Serial カテゴリより JTAG UART をダ ブルクリックします。こちらのウィザードもデフォルトの設定で Finish をクリックします。

🖕 JTAG UART - jtag_uart_0	
JTAG UART altera_avalon_itag_uart	Documentation
Block Diagram     Show signals     jtag_uart_0     ck     elock interrupt     reset     reset     avalon_itag_slave     avalon     atera_avalon_itag_uart	Write FIFO (Data from Avalon to JTAG)     Buffer depth (bytes): 64      IRQ threshold: 8     Construct using registers instead of memory blocks     Wread FIFO (Data from JTAG to Avalon)     Buffer depth (bytes): 64      IRQ threshold: 8     Construct using registers instead of memory blocks     Vallow multiple connections     Allow multiple connections to Avalon JTAG slave
	Cancel

⑥ 各コンポーネントを接続します。Connections 欄の白丸をクリックすることで下図のように黒丸に変えることで接続します。



 割り込みポートを接続します。IRQ 欄の JTAG\_UART の Interrupt Sender と Nios II Processor の Interrupt Receiver を下図のように接続します。値は優先順位を示しますが、ここでは、割り込みが1本な ので0固定となります。

🗆 nios2_qsys_0	Nios II Processor					
clk	Clock Input	Double-click to export	clk_0			
reset_n	Reset Input	Double-click to export	[clk]			
data_master	Avalon Memory Mapped Master	Double-click to export	[clk]			
instruction_master	Avalon Memory Mapped Master	Double-click to export	[clk]			
d_irq	Interrupt Receiver	Double-click to export	[clk]		IRQ 0	IRQ 31 ← \
jtag_debug_module_re	Reset Output	Double-click to export	[clk]			
jtag_debug_module	Avalon Memory Mapped Slave	Double-click to export	[clk]	■ 0x0000_0800	0×0000_0fff	
custom_instruction_m	Custom Instruction Master	Double-click to export				
🖯 pio_0	PIO (Parallel I/O)					
clk	Clock Input	Double-click to export	clk_0			
reset	Reset Input	Double-click to export	[clk]			
s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	₽° 0x0000_0000	0×0000_000f	
external_connection	Conduit	Double-click to export				
🗆 jtag_uart_0	JTAG UART					
clk	Clock Input	Double-click to export	clk_0			
reset	Reset Input	Double-click to export	[clk]			
avalon_jtag_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]		0×0000_0007	
irq	Interrupt Sender	Double-click to export	[clk]			

② Nios II の Reset Vector と Exception Vector の設定を行います。Qsys 上で Nios II Processor をダブル クリックし、再度 Nios II の設定画面を開きます。そこで、プルダウン・メニューからオンチップメモリを指 定します。以下のように設定します。

🔄 Parameters 🛛	
nios2_system > nios2_qsys_0	
Nios II Processor altera_nios2_qsys	
▼ Reset Vector	
Reset vector memory:	onchip_memory2_0s1
Reset vector offset:	0×00000000
Reset vector:	0×00020000
Exception Vector	
Exception vector memory	onchip_memory2_0s1
Exception vector offset:	0×0000020
Exception vector:	0×00020020
MMU and MPU	
🔲 Include MMU	

③ PIO の出力信号を Qsys の外部に Export します。Export 行にて external\_connection に該当するセ ルをダブルクリックし、Enter で確定させます。

						~ ```	custom_instruction_m	Oustom Instruction Master	Double-click to export	
$\checkmark$							🖾 pio_0	PIO (Parallel I/O)		
	♦—	$\left  \right $	_		$\neg$	$\rightarrow$	clk	Clock Input	Double-click to export	clk_0
	•	┝┤	_		-	$\rightarrow$	reset	Reset Input	Double-click to export	[clk]
			H	>	_	$\rightarrow$	s1	Avalon Memory Mapped Slave	Double click to experi	[clk]
						- <b>*</b> -	external_connection	Conduit	pio_0_external_connection	

ALTIMA

④ ベースアドレスを重複のないように設定します。System メニューの Assign Base Address を実行します。
 下図のように設定(値は、Quartus II のバージョンや設定順序により、変わる場合があります)され、
 Message 画面からエラーが消えるのが確認できます。

1	System	Contents 🕺 Address	Map 🖾 Interconnect Re	equirements 🛛 Device Family 🖇	3				- 🗗 🗆
+	Use	Connections	Name	Description	Export	Clock	Base	End	IRQ
×			≡ clk_0	Clock Source					
		D-	clk_in	Clock Input	clk	exported			
1. <b>1.</b>		·	clk_in_reset	Reset Input	reset				
X			clk	Clock Output	Double-click to export	clk_0			
			clk_reset	Reset Output	Double-click to export				
_			🗆 onchip_memory2_0	On-Chip Memory (RAM or ROM)					
		$\bullet$ $\rightarrow$	clk1	Clock Input	Double-click to export	clk_0			
		$     \bullet \bullet \longrightarrow$	s1	Avalon Memory Mapped Slave	Double-click to export	[clk1]	= 0x0002_0000	0×0003_ffff	
-		$   \bullet   \bullet \rightarrow$	reset1	Reset Input	Double-click to export	[clk1]			
Y			🗆 nios2_qsys_0	Nios II Processor					
		$  \bullet   + + + \rightarrow$	clk	Clock Input	Double-click to export	clk_0			
		$   \uparrow \uparrow \downarrow \uparrow \uparrow \rightarrow$	reset_n	Reset Input	Double-click to export	[clk]			
			data_master	Avalon Memory Mapped Master	Double-click to export	[clk]			
			instruction_master	Avalon Memory Mapped Master	Double-click to export	[clk]			
		$        \longrightarrow$	d_irq	Interrupt Receiver	Double-click to export	[clk]	IRQ 0	IRQ 31	
			jtag_debug_module_re	Reset Output	Double-click to export	[clk]			
		$     \bullet \bullet   \longrightarrow$	jtag_debug_module	Avalon Memory Mapped Slave	Double-click to export	[clk]	= 0x0004_0800	0×0004_0fff	
		×	custom_instruction_m	Custom Instruction Master	Double-click to export				
			🗆 pio_0	PIO (Parallel I/O)					
		$  \bullet     \rightarrow$	clk	Clock Input	Double-click to export	clk_0			
		$   \bullet   \bullet \to$	reset	Reset Input	Double-click to export	[clk]			
		$     \bullet \bullet   \longrightarrow$	s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	=° 0×0004_1000	0×0004_100f	
		· · ·	external_connection	Conduit	pio_0_external_connection				
			🖂 jtag_uart_0	JTAG UART					
		• • • • • • • • • • • • • • • • • • • •	clk	Clock Input	Double-click to export	clk_0			
		$\bullet \qquad \bullet \qquad$	reset	Reset Input	Double-click to export	[clk]			
		$\bullet \bullet \longrightarrow$	avalon_jtag_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	■ 0×0004_1010	0×0004_1017	
		•─────────────────────────────────	irq	Interrupt Sender	Double-click to export	[clk]			

 ⑥ システムを Generate します。Generate メニューから Generate HDL を選択します。下図の画面が表示 されるので、Generate ボタンをクリックします。Generate が終了したら、Generated Completed 画面の Close をクリックしてください。

Synthesis files are used to co	ompile the system in a Quartus II project.
Oreate HDL design files for s	ynthesis: Veril 👻
Create timing and resour	ce estimates for third-party EDA synthesis tools.
Create block symbol file	(bsf)
<u>Circulation</u>	
Simulation	
The simulation model contain	s generated HDL files for the simulator, and may include simulation-only features.
oreate simulation model.	None
Allow mixed-language si	nulation
Enable this if your simulator :	supports mixed-language simulation.
Output Directory	
Path:	C:/Lab/nios2 basic pri/nios2 system

- 2-4. Quartus II にてコンパイル
  - 今回はトップのファイルとして回路図 (.bdf) ファイルを採用しましたが、当然トップのファイルは H DL で 構いません。Quartus II の File メニューの New をクリックし、New ウインドウより Block Diagram/Schematic File を選択し OK をクリックします。



② 表示された Block1.bdf ファイル上でダブルクリックします。Symbol ウインドウが表示されますので、ファ イル選択ボタンをクリックして、Qsys が生成したフォルダ (nios2\_system) 内の nios2\_system.bsf ファイ ルを選択し、Open をクリックします。すると、Symbol ウインドウに該当の BSF ファイルが表示されるの で、OK ボタンをクリックして回路図上に反映させます。



③ 先程と同じように、B lock1.bdf ファイル上でダブルクリックします。Symbol ウインドウが表示されますので、その Name 欄で input と入力し OK をクリックします。入力ピンのシンボルが用意されるので、Qsys のシンボルに接続します。clk、reset\_n にそれぞれ、入力ピンのシンボルを接続します。同じように、Name 欄に output と入力することで、出力ピンのシンボルが用意されます。さらに同じように、Name 欄に not と入力することでインバータのシンボルが用意されるので、led のノードに接続しその外側へ出力ピンのシンボルを接続します。それぞれのシンボルはダブルクリックすることで名前の変更ができます。各シンボルの名前を次の画面の様に、clk、reset\_n、led\_pio[7..0] としてください。



- ④ File メニュー → Save As よりトップの .bdf ファイルの保存をします。ここでは、ファイル名を nios2\_basic\_lab.bdf としました。
- ⑤ Qsys が生成した QIP ファイルをプロジェクトに設定します。Assignment メニューの Settings を実行し、 Files カテゴリからファイル選択ボタンで nios2\_system\_synthesis/nios2\_system.qip を選択し、Add ボタン で確定させます。設定されたら OK ボタンで画面を終了します。

Settings - nios2_basic_lab			- • •
Category:			Device
General	Files		
Files Libraries	Select the design files you want to include in the project. Cli directory to the project.	ck Add All to add all design files ir	n the project
<ul> <li>IP Settings</li> <li>IP Catalog Search Locations</li> </ul>	Eile name:		Add
<ul> <li>Operating Settings and Conditions</li> </ul>	File Name	Туре	Add All
Voltage	nios2_system/synthesis/nios2_system.qip	IP Variation File (.qip)	Demove
Temperature	nios2_basic_lab.bdf	Block Diagram/Schemati	
<ul> <li>Compilation Process Settings</li> </ul>			<u>U</u> p
Incremental Compilation			Down
Physical Synthesis Optimizatio			
EDA Tool Settings			Properties

⑥ トップのデザインを Quartus II のチェックにかけます。

Quartus II  $\mathcal{O}$  Processing  $\checkmark = \neg \rightarrow$  Start Analysis & Elaboration  $\mathcal{E}\mathcal{O}\mathcal{I}\mathcal{I}\mathcal{I}\mathcal{I}\mathcal{I}\mathcal{I}$ 



 ⑦ ピン・アサインを行います。Quartus II の Assignments メニューの Pin Planner をクリックします。Pin Planner が表示されるので、次の表に対応するようにピン・アサインを行ってください。Terasic 社の Cyclone V GX スターター開発キット以外のボードを使用する場合は、ご使用のボード仕様を確認後、ピン配置の設定をしてください。

clk	R20
led_pio[7]	H9
led_pio[6]	H8
led_pio[5]	B6
led_pio[4]	A5
led_pio[3]	E9
led_pio[2]	D8
led_pio[1]	K6
led_pio[0]	L7
reset n	AB24

⑧ Location 欄へカーソルを持っていき、ピン・アサインを行いたい番号を入力、もしくはプルダウン・メニューより選択してください。

Node Name	Direction	Location	I/O Bank	
i≞_ clk	Input	PIN_R20	5B	
ut led_pio[7]	Output	PIN_H9	8A	
et led_pio[6]	Output	PIN_H8	8A	
et led_pio[5]	Output	PIN_B6	8A	
out led_pio[4]	Output	PIN_A5	8A	
et led_pio[3]	Output	PIN_E9	8A	
et led_pio[2]	Output	PIN_D8	8A	
et led_pio[1]	Output	PIN_K6	8A	
et led_pio[0]	Output	PIN_L7	8A	
🖳 reset_n	Input	PIN_AB24	5A	
< <new node="">&gt;</new>				

9 Quartus II にてトップのデザインをコンパイルします。

Processing  $\forall = a - \rightarrow$  Start Compilation  $\epsilon p = 0$ 

ハー ドウェアの生成は以上です。

※本来は、SDC ファイルを生成し、タイミング制約を行う必要がありますが、本資料では省略しています。

\Lambda ALTIMA

2-5. システムを FPGA ヘダウンロード

 USB-Blaster を接続し、ボードに電源を供給した後、Quartus II の Programmer より SOF ファイルを FPGA にダウンロードします。Tools メニューより Programmer をクリックします。SOF ファイルを選択し、 Program/Configure のチェックを入れて Start をクリックします。

👾 Programmer - C:/	/Lab/nios2_basic_prj/nio	s2_basic_lab - nios	2_basic_lab -	[Chain2.cdf]*						-	
<u>File Edit V</u> iew	P <u>r</u> ocessing <u>T</u> ools <u>W</u> i	ndow <u>H</u> elp 🕏							Search	altera.co	m 🚯
🔔 Hardware Setup	USB-Blaster [USB-0]			Mode	; JTAG		•	Progress:	100%	(Succes	iful)
Enable real-time ISP	to allow background program	ming (for MAX II and M	1AX V devices)								
Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine	Security Bit	Erase	ISP CLAMP
Stop	output_files/nios2_b…	5CGXFC5C6F27	03790D51	03790D51	V						
Auto Detect											
🔀 Delete											
Add File											
隆 Change File	•										
Save File											
Add Device											
1 1 Uo											=
4 a Down											
	5CGXFC5C6	■_  =27									-
		1									

#### 3. <u>ソフトウェア</u>

- 3-1. Nios II SBT でソフトウェア・プロジェクトの作成
  - Windows のスタートメニューより Nios II SBT を開きます。Workspace Launcher 画面が起動するので、 任意のフォルダを指定します。ここでは、予め software という名前のフォルダを作成し、そこをワークス ペースとして指定しています。OK をクリックし、Nios II SBT を起動します。

Workspace Launcher	×
Select a workspace	
Eclipse stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
<u>W</u> orkspace: C:¥Lab¥nios2_basic_prj¥software	
Use this as the default and do not ask again	
OK Cancel	

② Nios II SBT でソフトウェアのプロジェクトを作成します。

Nios II SBT の File メニュー → New → Nios II Application and BSP from Template をクリックします。 Nios II Application and BSP from Template 画面が起動します。

- ③ SOPC Information File name に、SOPCINFO ファイルを選択します。これは、Qsys 生成時に作成された ファイルで、Qsys 内のシステム情報をソフトウェアに渡すために参照されるファイルとなります。
- ④ Project name に任意のプロジェクト名を入力します。ここでは、soft\_test と入力します。
- ⑤ 最後に Project Template 欄の Templates で、Blank Project を選択し、Finish をクリックします。

Nios II Application and BSP from Template	- • •
Nios II Software Examples Create a new application and board support package based on a software example template	
Target hardware information         SOPC Information File name:         C:¥Lab¥nios2_basic_prj¥nios2_system.sopcinfo         CPU name:         nios2_qsys_0	
Application project Project name: soft_test  Vest default location Project location: C:¥Lab¥nios2_basic_prj¥software¥soft_test  Project template  Template description Blank Project to which you Count Binary Float2 Functionality Float2 Functionality Float2 CCC Float2 Performance Hello Freestanding Hello MicroC/OS-II	
(?) < Back Next > Einish	Cancel

⑥ Nios II のソフトウェアを記述します。Nios II SBT の File メニュー → New → Other... → C/C++ → Source File を選択し、Next をクリックします。New Source File ウインドウの、Source Folder 欄に、Browse ボタンよりソフトウェア・プロジェクトのアプリケーション・プロジェクト(\_bsp がついていない方のフォルダ)を選択します。Source File 欄には、任意のソフトウェアのファイル名を拡張子.c をつけて入力し、Finish をクリックします。ここでは、soft\_test.c と入力します。

New Source File	- • •
Source File Create a new source file.	C
Source folder: soft_test	Browse
Source file: soft_test.c	
Template: Default C++ source template	Configure
Image: Second	Cancel

⑦ Nios II SBT に新規でファイルが用意されたので、そちらにソフトウェアのコードを記述します。以下の記述を行ってください。

<pre>#include <stdio.h> #include <unistd.h> #include "system.h" #include "altera_avalon_pio_regs.h"</unistd.h></stdio.h></pre>
int min()
int main()
<pre>print+( "Hello from Nios II! Start!\n");</pre>
while(1)
<pre>IOWR_ALTERA_AVALON_PIO_DATA(PIO_0_BASE, 0x55); usleep(500000);</pre>
IOWR_ALTERA_AVALON_PIO_DATA(PIO_0_BASE, 0xAA); usleep(500000);
}
return 0;
}

⑧ システムの設定を確認します。soft\_test\_bsp フォルダを右クリックし、Nios II → BSP Editor... を実行すると Nios II BSP Editor が起動します。Linker Script タブを指定し、プログラム・メモリに、onchip\_memory2\_0 が設定されていることが確認できます。Generate をクリックして Exit で閉じます。

in NIOS II BSP Editor - se	ettings.bsp					
e Edit Tools Help			1			
Iain Software Packages D	rive s Linker Script ina	able File Generation	Target BSP Directory			
Linker Section Mappings						
Linker Section Name		Linker Region Na	me	Memory Device Name		Add
.bss		onchip_memory2	_0	onchip_memory2_0		Remove
.entry		reset		onchip_memory2_0		Restore Defaults
exceptions		onchip_memory2	_0	onchip_memory2_0		-
.heap		onchip_memory2	_0	onchip_memory2_0		
.rodata		onchip_memory2	_0	onchip_memory2_0		
.rwdata		onchip_memory2	_0	onchip_memory2_0		
.stack		onchip_memory2	_0	onchip_memory2_0		
.text		onchip memory2	0	onchip memory2 0		
Linker Memory Regions						
Linker Degion Name	Address Dange	*	Memory Dovice Name	Size (hutee)	Offrat (hutaa)	Add
unker Kegion Hame	0.00020020	- 0.00095555	Intention y Device Name	1910.40	onacc (bytea)	Remove
onchip_memoryz_u	0x00020020	- 0x0003FFFF	onchip_memory2_0	131040	0	2 Dectars Definition
						Remove Memory Device Memory Usage Memory Map
rayed out entries are auto	omatically created at ge	enerate time. The	y are not editable or persiste	d in the BSP settings file.		
nformation Problems Proc	essing					
Setting "hal.linker.interrup	t_stack_memory_region_r	name" set to "onchi	p_memory2_0".			
Setting "hal.linker.exception	on_stack_memory_region	_name" set to "onch	nip_memory2_0".			
Loading drivers from ense	mble report.					
Mapped module: "nios2_g	sys_0" to use the default of	driver version.				
Mapped module: "pio 0" to	use the default driver ve	ersion.				
Mapped module: "itan uar	t 0" to use the default dr	iver version.				
	om ensemble report.					
Finished loading drivers from the second	settings file.					
Finished loading drivers from Loading BSP settings from	and the second sec					
Finished loading drivers from Loading BSP settings from Finished loading SOPC Built	der system infn file "\\	nine? system conci	nfo [relative to settings file]"			
<ul> <li>Finished loading drivers fro</li> <li>Loading BSP settings from</li> <li>Finished loading SOPC Buil</li> </ul>	der system info file "\\	nios2_system.sopci	nfo [relative to settings file]"			

⑨ ソフトウェアをビルドします。Nios II SBT 左枠のアプリケーション・プロジェクトのフォルダ (\_bsp がつい ていない方)をハイライトし、右クリック → Build Project をクリックします。

🎦 Project	Explo	orer 🕱 🗧	🛯 🗖 🚺 soft_test.c 🛛	
C coft	toot	E 🕏 🕯	<pre></pre>	
≥ sort ⊳ 🔊 I ⊳ 💽 s		New Go Into	•	
	-	Open in New Windo	w	
		Сору	Ctrl+C	
⊳ 1≅ soft	Ê	Paste	Ctrl+V	
	×	Delete	Delete	.cegs.h"
	<u>.</u>	Remove from Conte	ext Ctrl+Alt+Shift+Down	
		Source	+	
		Move		ş II! Sta
		Rename	F2	
	2	Import		PIO DATA(
	4	Export		
		Build Project		PIO_DATA(I
		Clean Project		
	\$	Refresh	F5	
		Close Project		

3-2. 実行コードをターゲットで実行、デバッグ

① ビルドしたソフトウェアをターゲットのシステムで実行します。Nios II SBT 左枠のアプリケーション・プロジェクトのフォルダ(\_bsp がついていない方)をハイライトし、右クリック  $\rightarrow$  Run As  $\rightarrow$  Nios II Hardware を クリックします。

Build Configurations Make Targets Index	Properties
Show in Remote Systems view Convert To	program size (code + initialized data) free for stack + heap.
Run As	🕨 🌲 1 Lauterbach ISS 🛛 est
Debug As	2 Local C/C++ Application
Profile As	3 Nios II Hardware
Team	<ul> <li>4 Nios II Hardware v2 (beta)</li> </ul>
Compare With	5 Nios II ModelSim
Restore from Local History Nios II	Run Configurations
	Build Configurations Make Targets Index Show in Remote Systems view Convert To Run As Debug As Profile As Team Compare With Restore from Local History Nios II

 Nios II Console ウインドウに、printf の出力のキャラクタが確認できます。また、ターゲット上の LED の 点灯がプログラムの通りの動作になっていることを確認します。

	P			
🖹 Problems 🖉 Tasks 🗐 Console 🗇 Properties 🛗 Nios II Console 🕱			🔳 🛃 🚉 🛯	
soft_test Nios II Hardware configuration - cable: USB-Blaster on localhost [USB-0] device ID: 1 instance	D: 0 name	: jtaguart_0		
Hello from Nios II! Start!				

③ デバッガを起動します。Nios II SBT 左枠のアプリケーション・プロジェクトのフォルダ (\_bsp がついていない方)を ハイライトし、右クリック  $\rightarrow$  Debug As  $\rightarrow$  Nios II Hardware をクリックします。

Build Configurations     Index			
Show in Remote Systems view Convert To Run As			
Debug As	C	1 Local C/C++ Application	-
Profile As	2	2 Nios II Hardware	
Team	2	3 Nios II Hardware v2 (beta)	
Compare With Restore from Local History		Debug Configurations	

④ デバッグ・ウインドウを開くために以下のウインドウで、Yes をクリックします。

Confirm	m Perspective Switch
?	This kind of launch is configured to open the Nios II Debug perspective when it suspends.
	This perspective is designed to support efficient development of a Nios II project. It turns off automatic build and adds the Nios II Console view which is particularly useful.
	Do you want to open this perspective now?
<u>R</u> em	nember my decision
	Yes No

⑤ Nios II SBT のウインドウがデバッグ・ウィンドウに切り替わり、デバッグが行えるようになりました。ここで、 ソフトウェア・ブレークポイントやステップ実行等が行うことができます。ブレークポイントは、プログラム・ ソースコードが表示されているウインドウの左端をダブルクリックすることにより設定します。ブレークポイ ントを設定した箇所にはアイコンが表示されます(下画面赤枠をご参照ください)。設定を解除する場合は、 このブレークポイントのアイコンをダブルクリックします。

Nios II Debug - soft_test/soft_test.c - Eclipse Eile Edit Source Refector Navigate Search Run Project Nios II Window	r Haln				
		🗄 🖗 Nios II Deb 👔			
🌾 Debug 🛛 💫 🍇 🖉 🕪 🗉 🔳 🖉 🔁 🖧 🤯 🖄 👘 😵 🖉	🗱 Variables 🛛 💊 Breakpoints 👯 Re	egisters 🚺 Memory			
soft_test Nios II Hardware configuration [Nios II Hardware]		Ё ಈ ⊑ 🗳 ¥ 🔌 🖸 ೮ ▽			
Altera CDI GDB Debugger (14/07/31 16:20) (Suspended)	Name	Value			
= 1 main() soft test c:17 0x00020214					
rman() sol_testerr) 0x00020214 sol_testerr) 0x00020214 sol_testerr) 0x00020214					
nios2-download (14/07/31 16:20)					
📕 nios2-elf-gdb (14/07/31 16:20)		A			
		<b>v</b>			
		4			
c soft_test.c ☆		E Outline 🛛 🗌			
<pre>#include "altera_avalon_pio_regs.h"</pre>	<b>^</b>	ja †s x x a # ∠			
int main()		stdio.h			
<pre>printf( "Hello from Nios II! Start!\n");</pre>	unistd.h				
		system.n			
		main() : int			
IOWR_ALTERA_AVALON_PIO_DATA(PIO_0_BASE, 0x55); usleep(500000):					
IOWR ALTERA AVALON PIO DATA(PIO @ BASE, @xAA);	IOWR ALTERA AVALON PIO DATA(PIO @ BASE, @XAA);				
usleep(500000);					
I					
return 0;	<b>v</b>				
soft test Nins II Hardware configuration - cable: USB-Blaster on localhost (USB-0) device ID: 1 instance	D: 0 name: itaquart 0				
□ <sup>◆</sup> Writable Smart Insert					

⑥ デバッガのアイコンの情報は以下になります。



#### <u> 改版履歴</u>

Revision	年月	概要
1	2014 年 8 月	新規作成

#### 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。