

Nios[®] II はじめてガイド Nios[®] II 簡易チュートリアル

Ver.17



2017 年 11 月 Rev.1

ELSENA,Inc.





Nios® II はじめてガイド

Nios[®] II 簡易チュートリアル

<u>目次</u>

1.	はじめに	3
2.	ハードウェア	5
Ĩ	2-1. ハードウェア・プロジェクトの作成	. 5
Ĩ	2-2. コンフィギュレーション・モードの設定	. 6
Ĩ	2-3. Qsys にてシステムを構成し、HDL を生成	. 7
Ĩ	2-4. Quartus [®] Prime でコンパイル	12
2	2-5. システムを FPGA ヘダウンロード	15
3.	ソフトウェア	16
3	3-1. Nios® II SBT でソフトウェア・プロジェクトの作成	16
3	3-2. 実行コードをターゲットで実行、デバッグ	20
改	版履歴	24



1. <u>はじめに</u>

この「Nios®IIはじめてガイド」シリーズは、Nios®IIプロセッサをはじめて使用するユーザ向けの資料です。

この資料は、インテル[®] Quartus[®] Prime 開発ソフトウェアや Nios[®] II Software Build Tools for Eclipse (Nios[®] II SBT)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とソフトウェ 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理解することができます。

この資料で説明に使用するデザインのハードウェアの構成は、Nios[®] II コアや 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。

【開発ソフトウェア】

- インテル[®] Quartus[®] Prime 開発ソフトウェア 17.0
- Nios[®] II Software Build Tools for Eclipse (Quartus Prime 17.0) ※ 以降、Nios[®] II SBT と記載

【フローの概要】

- ① Quartus[®] Prime の起動
- ② Qsys システム統合ツールにて、Nios® II システムの構築、生成
- ③ Quartus[®] Prime 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 スタータ開発キットを使用して説明します。他の評価ボードなどを 使用する場合は、Quartus[®] Prime にて設定するデバイス型名の選択やピン配置、未使用ピンの処理などをそれぞ れのボードの仕様に合わせる必要があります。





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
- ① Quartus[®] Prime を起動して、File メニュー ⇒ New Project Wizard を選択します。New Project Wizard の Introduction ウィンドウが表示されたら Next を 1 回クリックして、以下の画面でワーキング・ディレクトリとハー 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。

S New Project Wizard	
Directory, Name, Top-Level Entity	
What is the working directory for this project?	
C:\Lab\nios2_basic_prj	
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 e design file.	ntity name in the
nios2_basic_lab	
Use Existing Project Settings	
< <u>Back</u> <u>Next></u> <u>Einish</u> Ca	incel <u>H</u> elp

 ② Next を 3 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、Cyclone[®] V を選択し、 Available devices 中の 5CGXFC5C6F27C7 を選択します。Name filter にデバイスの型番を入力します。

elect the family and ou can install additi	device you want to tai onal device support w	rget for comp ith the Install	llation. Devices command	d on the Tool	ls menu.				
o determine the ver	sion of the Quartus Pri	ime software i	n which your targ	et device is s	supported, refer t	o the <u>Device</u>	Support List webpage.		
Device family				Sh	now in 'Available	devices' list			
Eamily: Cyclone V	(E/GX/GT/SX/SE/ST)			* Pa	ackage:	Any		•	
Device: All	Device: All			Pi	Pin <u>c</u> ount:		Any		
Target device	arget device				Core speed grade:		Any		
Auto device sel	ected by the Fitter			N	Name filter: 5CGX		5CGXFC5C6F27C7		
Specific device	selected in 'Available d	levices' list		1	Show advanced devices				
Other: n/a									
vailable devices:									
	Core Voltage	ALMs	Total I/Os	GPIOs	GXB Chan	nel PMA	GXB Channel PCS	PCIe Hard IP Bl	
Name								14 65 5	

 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 でこのウィンドウを閉じます。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 Quartus[®] Prime の Assignments メニュー \Rightarrow Device を選択し、Device and Pin Options をクリックします。

elect the family and o	levice you want to ta	rget for comm	ilation			
ou can install additio	nal device support w	ith the Install	Devices comman	d on the Tools	menu.	
o determine the versi	ion of the Quartur Bri	merceftware	in which your tar	tot davica is su	posted refer to the David	a Support List webpo
o determine the vers	ion of the Quartus Ph	ine sortware	in which your targ	get device is su	pported, refer to the <u>bevic</u>	<u>Le Support List</u> webpa
Device family			Sh	now in 'Availabl	le devices' list	
Eamily: Cyclone V	(E/GX/GT/SX/SE/ST)		▼ Pa	ac <u>k</u> age:	Any	•
Device: All			▼ Pi	n count:	Any	•
-						-
Target device			C	ore sp <u>e</u> ed grad	e: Any	· ·
Auto device sele	cted by the Fitter		Na	ame filter:		
				-		
Specific device s	elected in 'Available c	levices' list		Show advanc	ed devices	
<u>Specific device s</u> Other p/a	elected in 'Available c	levices' list		S <u>h</u> ow advanc	ed devices	
 Specific device s Other: n/a 	elected in 'Available c	levices' list	Dev	S <u>h</u> ow advanc	ed devices	
Specific device s Other: n/a	elected in 'Available c	levices' list		Show advanc	ed devices	
Specific devices Qther: n/a vailable devices: Name	elected in 'Available o Core Voltage	levices' list ALMs	Total I/Os	Show advance vice and Pin Op GPIOs	ed devices btions GXB Channel PMA	GXB Channel P
Specific device s Other: n/a vailable devices: Name SCGXFC5C6F23A7	elected in 'Available of Core Voltage	ALMs 29080	Total I/Os 268	Show advance vice and Pin Op GPIOs 240	GXB Channel PMA	GXB Channel P
Specific device s Qther: n/a vailable devices: Name SCGXFC5C6F23A7 SCGXFC5C6F23C6	elected in 'Available o Core Voltage 1.1V 1.1V	ALMs 29080 29080	Des Total I/Os 268 268	Show advance vice and Pin Op GPIOs 240 240	GXB Channel PMA	GXB Channel P 6 6
Specific device s Qther: n/a vailable devices: Name SCGXFC5C6F23A7 SCGXFC5C6F23C6 SCGXFC5C6F23C7	Core Voltage 1.1V 1.1V 1.1V 1.1V	ALMs 29080 29080 29080	Total I/Os 268 268 268 268	Show advance vice and Pin Op GPIOs 240 240 240	GXB Channel PMA	GXB Channel P 6 6 6
Specific device s Qther: n/a quilable devices: Name SCGXFC5C6F23A7 SCGXFC5C6F23C6 SCGXFC5C6F23C7 SCGXFC5C6F23I7	Core Voltage 1.1V 1.1V 1.1V 1.1V 1.1V 1.1V	ALMs 29080 29080 29080 29080 29080	Total I/Os 268 268 268 268 268	Show advance Show advance S	GXB Channel PMA 6 6 6 6 6 6	GXB Channel P 6 6 6 6 6
Specific device s Qther: n/a galable devices: Name SCGXFC5C6F23A7 SCGXFC5C6F23C6 SCGXFC5C6F23C7 SCGXFC5C6F23C7 SCGXFC5C6F23C7	Core Voltage 1.1V 1.1V 1.1V 1.1V 1.1V 1.1V 1.1V	ALMs 29080 29080 29080 29080 29080 29080 29080	Total I/Os 268 268 268 268 268 268 364	S <u>h</u> ow advance vic∈ and Pin Op 240 240 240 240 240 336	GXB Channel PMA 6 6 6 6 6 6 6 6	GXB Channel P 6 6 6 6 6 6
Specific device s Qther: n/a Qther: n/a valiable devices: Name SCGXFC5C6F23A7 SCGXFC5C6F23C6 SCGXFC5C6F23C7 SCGXFC5C6F22C7 SCGXFC5C6F27C7	Core Voltage 1.1V 1.1V 1.1V 1.1V 1.1V 1.1V 1.1V 1.1V	ALMs 29080 29080 29080 29080 29080 29080 29080 29080	Total I/Os 268 268 268 268 268 364 364	Show advance GPIOs 240 240 240 336 336	GXB Channel PMA 6 6 6 6 6 6 6 6 6 6	GXB Channel P 6 6 6 6 6 6 6
Specific device s Qther: n/a yallable devices: Name SCGXFC5C6F23A7 SCGXFC5C6F23C6 SCGXFC5C6F2376 SCGXFC5C6F2776 SCGXFC5C6F2776	Core Voltage 1.1V 1.1V	ALMs 29080 29080 29080 29080 29080 29080 29080 29080 29080 29080 29080	Total I/Os 268 268 268 268 268 364 364 364	Show advance GPIOs 240 240 240 336 336 336 336	GXB Channel PMA 6 6 6 6 6 6 6 6 6	GXB Channel P 6 6 6 6 6 6 6 6 6
Specific device s Qther: n/a yallable devices: Name SCGXFC5CGF23A7 SCGXFC5CGF23C7 SCGXFC5CGF23C7 SCGXFC5CGF23C7 SCGXFC5CGF27C7 SCGXFC5CGF2777 4	Core Voltage 1.1V 1.1V 1.1V 1.1V 1.1V 1.1V 1.1V 1.1	ALMs 29080 29080 29080 29080 29080 29080 29080 29080	Total I/Os 268 268 268 268 268 364 364 364 364	Show advance vice and Pin Op 240 240 240 336 336 336	ded devices	GXB Channel P 6 6 6 6 6 6 6 6 6

 Device and Pin Options ウィンドウの Configuration を開き、Configuration scheme で Active Serial x1(もしくは、 Active Serial x4)を選択し、OK をクリックします。

1	Configuration					
eneral	Comparation					
rogramming Files	Specify the device configuration scheme	and the configuration device.				
Inused Pins	Configuration scheme: Active Serial x1	(can use Configuration Device)				
Jual-Purpose Pins						
apacitive Loading	Configuration mode: Standard					
loard Trace Model	Configuration device					
/O Timing	Aut	0				
/oltage	☑ Use configuration device:					
Pin Placement		Configuration Device Options				
irror Detection CRC	Configuration device I/O voltage: Aut	0				
artial Reconfiguration	Force VCCIO to be compatible with	Earce VCCIO to be compatible with configuration I/O voltage				
and a recomputation						
	VID Operation mode					
	Configuration pin:	Configuration Pin Ontions				
	Generate compressed bitstreams					
	Active serial clock source: 100 MHz Inte	ernal Oscillator				
	Enable input tri-state on active config	guration pins in user mode				
	Description:					
	The method used to configure a device	with a design. Available configuration schemes				
	The method used to configure a device with a design. Available configuration schemes depend on selected device family: Passive Serial (PS), Passive Parallel x8 (PPx8), Passive					
	Parallel x16 (PPx16), Passive Parallel x3	2 (PPx32), Active Serial x1 (ASx1), Active Serial x4				
	(ASx4) and AVST x8, x16 and x32.					
		Reset				

2-3. Qsys にてシステムを構成し、HDL を生成

① Quartus[®] Prime の Tools メニュー ⇒ Qsys を選択し、Qsys を開きます。Qsys の File メニュー ⇒ Save as を 選択して、先にファイルをセーブします。今回は、nios2_system.qsys というファイル名で保存します。



 ② 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) ウィンドウ内の Total memory size を 128 K Bytes (128k と入力で反映)に設定し Finish をクリックします。

•	- 14					
ck Diagram	Memory type					
ow signals	Туре	RAM (Writable) -				
and in an and a	Dual-port access					
onchip_memory2_0	Single clock operation					
- deat	Read During Write Mode:	DONT CAR				
0.000	Block type:	AUT -				
avalon		Hora •				
et1 reset						
era_avalon_onchip_memory2	" Size					
	Enable different width for Dual-p	int access				
	Slave S1 Data width:	32 -				
	Total memory size:	131072 bytes				
	Minimize memory block usage (m	ay impact (max)				
	In Development					
	Slave s1 Latency					
	Slave s2 Latency					
	ROM/RAM Memory Protection					
	Reset Request	Enabled •				
	* ECC Parameter					
	Extend the data width to support ECI	bits Disabl. 👻				
	* Memory initialization					
	Ditialize memory content					
	Enable considerault initialization :					
	Type the filename (e.g. my rai	inex) or select the nex file using the file prowser button.				
	Oser created initialization me.	onchip_mem.hex				
	Enable Partial Reconfiguration Initialization Mode					
	Enable In-System Memory Conte	t Editor feature				
	Instance ID	NONE				
	Memory will be initialized from	nios2_system_onchip_memory2_0.hex				
	1					

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。この時点で、Message ウィンドウにエラーが出ますが、気にせず先に進んでください。

Shork Diagram	Main Vectors Select an I Nios II Core:	Caches and Memory Interfaces Arithmetic mplementation Nice IV Nice IV Nice IV	Instructions MMU and MPU Settings JTAG Det	xue Advanced Features
reset avalon instruction_master		Nios II/e	Nios II/f	
inginterrupt resetdebug_reset_request	Summary	Resource-optimized 32-bit RISC	Performance-optimized 32-bit RISC	
debug_mem_elaveavalonnos_oution_instructioncustom_instructionattera_nos2_gen2	Features	JTAG Debug ECC RAM Protection	JTAG Debug Hardware Multiply/Divide Instruction/Data Caches Tighty-Coupled Masters ECC RAM Protection External Interrupt Controller Shadow Register Sets MPU MMU	E
	RAM Usage	2 + Options	2 + Options	
Error nios2_gen2_0 : Instruction Cache is larger than the Instruction Address. Please reduce the In: Error nios2 gen2 0 : Reset slave is not specified. Please select the reset slave	struction Cache S	Size. Current Tag Size is 0		

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 をクリックします。

5 PIO (Parallel I/O) - pio_0	
PIO (Parallel I/O)	
MegeCore altera_avalor_bio	
* Block Diagram	Basic Settings
Show signals	Width (1-32 bits):
pio 0	Direction: O Bidir
	Input
clock	🔘 InOut
reset	Output
avalon	Output Port Reset Value: 0x00000000000000
external_connection conduit	V Output Register
altera_avalon_pio	Enable individual bit setting/clearing
	Edge capture register
	Synchronously capture
	Edge Type: RISING
	Enable bit-clearing for edge capture register
	Thterrupt
	Generate IRQ
	IRQ Type:
	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
	Test bench wiring
	Drive inputs to field; Inventionantinon
	(د ۱
	Cancel

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。

Show signals jtag_uart_0 cik olock reset reset avalon_itag_slave avalon aitera_avalon_itag_uart	Write Flore (bytes): 64 IRQ threshold: 8 Read FIFO (Data from JTAG to Avalon) Buffer depth (bytes): 64 IRQ threshold: 8 Construct using registers instead of memory blocks Construct using registers instead of memory blocks
---	--

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 状態となります。



⑦ 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。ここの数字は割り込みの優先順位を表しており、数字が小さいほど優先度が高くなり ます。今回は割り込みが 1 本なので、0 となっています。

	日喧 nios2_gen2_0	Nios II Processor					
$ \qquad \qquad$	clk	Clock Input	Double-click to export	clk_0			
$ + + \rightarrow$	reset	Reset Input	Double-click to export	[clk]			
	data_master	Avalon Memory Mapped Master	Double-click to export	[clk]			
$ \rightarrow +$	instruction_master	Avalon Memory Mapped Master	Double-click to export	[clk]			
$ \longrightarrow$	irq	Interrupt Receiver	Double-click to export	[clk]		IRQ 0	IRQ 31
$ \rightarrowtail$	debug_reset_request	Reset Output	Double-click to export	[clk]			
$ \bullet \bullet \longrightarrow$	debug_mem_slave	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)					
\rightarrow	clk	Clock Input	Double-click to export	clk_0			
 $ \bullet \to \to$	reset	Reset Input	Double-click to export	[clk]			
$ \bullet \circ \longrightarrow$	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					
$\bullet + + - \rightarrow$	clk	Clock Input	Double-click to export	clk_0			
$\bullet + + \bullet \longrightarrow$	reset	Reset Input	Double-click to export	[clk]			
• • · · · · · · · · · · · · · · · · ·	avalon_jtag_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0000_0000	0×0000_0007	
•	irq	Interrupt Sender	Double-click to export	[clk]			

⑧ 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。Vectors タブをクリックします。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。本資料では、Nios®IIをオンチップ・メモリからブートする手法を説明 します。

is II Processor a_nios2_gen2	
in Vectors Caches and Memor	y Interfaces Arithmetic Instructions MMU and MPU Settings JTAG Debug Advanced Feature
leset vector memory.	onchip_memory2_0s1
Reset vector offset:	0×0000000
Reset vector:	0×00000000
Exception Vector	
xception vector memory	onchip_memory2_0s1 👻
exception vector offset:	0×00000020
xception vector:	0×00000020
Fast TLB Miss Exception Ve	ector
ast TLB Miss Exception vector n	nemory. None
ast TLB Miss Exception vector o	iffset 0x00000000
ast TLB Miss Exception vector:	0×0000000

 ⑨ 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。



① 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[®] Prime 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。File メニュー ⇒ Save で、Qsys システムを保存します。

🖾 Syste	m Contents 🛛 🛛 Address	Map 🛛 Interconnect Require	ments 🛛 Device Family 🖾					
	 System: nios2_syst 	em Path:clk_0						
+ Use	Connections	Name	Description	Export	Clock	Base	End	IRQ
1. V		⊟ clk_0	Clock Source					
×		≻ clk_in	Clock Input	clk	exported			
2	<u>ү</u> с	≻ clk_in_reset	Reset Input	reset				
		≺ clk	Clock Output	Double-click to export	clk_0			
A		≺ clk_reset	Reset Output	Double-click to export				
▲ 🗸		onchip_memory2_0	On-Chip Memory (RAM or ROM)					
•	•	→ clk1	Clock Input	Double-click to export	clk_0			
T		→ s1	Avalon Memory Mapped Slave	Double-click to export	[clk1]	■ 0x0002_0000	0x0003_ffff	
		→ reset1	Reset Input	Double-click to export	[clk1]	102		
		🗆 🖳 nios2_gen2_0	Nios II Processor					
	+ 	→ clk	Clock Input	Double-click to export	clk_0			
		→ reset	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]			
		→ irq	Interrupt Receiver	Double-click to export	[clk]	1	RQ C	IRQ 31 ←
		≺ debug_reset_request	Reset Output	Double-click to export	[clk]			
		→ debug_mem_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	<i>■</i> 0x0004_0800	0×0004_0fff	
		custom_instruction_master	Custom Instruction Master	Double-click to export				
V		🖯 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]	🛋 0x0004_1000	0×0004_100f	
	~	> external_connection	Conduit	pio_0_external_connection				
V		🗆 jtag_uart_0	JTAG UART					
	• • • • • • • • • • • • • • • • • • • •	→ clk	Clock Input	Double-click to export	clk_0			
	│	→ reset	Reset Input	Double-click to export	[clk]			
	• •	→ avalon_itag_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	■ 0×0004_1010	0×0004_1017	
	↓ •	≺ irq	Interrupt Sender	Double-click to export	[clk]	N		⊢(

 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。

Generation	
Y Sundhanin	
Synthesis	
Synthesis files are used to com	aplie the system in a Quartus Prime project.
Create HDL design files for syr	ithesis: Veril 👻
Create timing and resource	estimates for third-party EDA synthesis tools.
V Create block symbol file (b	nsf)
Simulation	
The simulation model contains	generated HDL files for the simulator, and may include simulation-only features.
Simulation scripts for this com	ponent will be generated in a vendor-specific sub-directory in the specified output directory
and ip-make-simscript comman	arated simulation scripts about how to structure your design's simulation scripts and how to use the <i>ip-setup-simulation</i> nd-line utilities to compile all of the files needed for simulating all of the IP in your design.
Create simulation model:	None -
* Output Directory	
Path:	C:/Lab/nios2_basic_prj/nios2_system

- 2-4. Quartus[®] Prime でコンパイル
- 今回はトップのファイルとして回路図(.bdf ファイル)を採用しますが、トップのファイルは HDL でも結構です。 Quartus[®] Prime の File メニュー ⇒ New を選択し、New ウィンドウより 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 ウィ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 ウィンドウに該当の 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。

Symbol		
Libraries:		
D 🖒 d./intelfpga/16.1/quartus/librar	Look in	ty Comp is protected by the sis is pr
Viane Bepeat-Inset mode Inset symbol as block		
es: d:/intelfpga/16.1/quartus/libraries/	File gar nios2_system	ne: nios2_system.bsf
system m psk-insert mode ert symbol as block e	ck ck pio_0_estemal_connection 0_external_connection_scont7_lowport reset_reset_n	

③ 先程と同じように、Block1.bdf ファイル上でダブルクリックします。Symbol ウィ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 Assignment メニュー ⇒ Settings を選択し、Files カテゴリで Qsys が生成した QIP ファイルがプロジェクトに登録 されているかを確認します。QIP ファイルがない場合は、ファイル選択ボタンで nios2_system_synthesis/nios2_system.qip を選択し、Add ボタンをクリックします。設定されたら OK ボタンで画面 を終了します。

tegory:			Device/Board
General	Files		
Files Libraries	Select the design files you want to in directory to the project.	clude in the project. Click Add All to add all design file	es in the project
IP Settings IP Catalog Search Locations	<u>F</u> ile name:		Add
Design Templates	۹.		X Add All
Voltage	File Name	Туре	Remove
Temperature	nios2_system/synthesis/nios2_	system.qip IP Variation File (.qip)	
Compilation Process Settings	nios2_basic_lab.bdf	Block Diagram/Schematic File	<u>p</u>
Incremental Compilation			Down
EDA Tool Settings Design Entry/Synthesis			Properties

⑥ Quartus[®] Prime の Processing メニュー ⇒ Start Analysis & Elaboration で、デザインを Quartus[®] Prime のチェック にかけます。



⑦ ピン・アサインを行います。Quartus[®] Prime の Assignments メニュー ⇒ 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 開発キット以外のボードを使用する場合は、使用する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・メニュー選択して ください。

Node Name	Direction	Location	I/O Bank
in_ clk	Input	PIN_R20	5B
ed_pio[7]	Output	PIN_H9	8A
bio[6]	Output	PIN_H8	8A
ed_pio[5]	Output	PIN_B6	8A
bio[4]	Output	PIN_A5	8A
Led_pio[3]	Output	PIN_E9	8A
ed_pio[2]	Output	PIN_D8	8A
out led_pio[1]	Output	PIN_K6	8A
ed_pio[0]	Output	PIN_L7	8A
in_ reset_n	Input	PIN_AB24	5A
< <new node="">></new>	1.1		-

- ⑨ Quartus[®] Prime の Processing メニュー ⇒ Start Compilation を選択して、トップのデザインをコンパイルします。
 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

インテル[®] FPGA ダウンロード・ケーブル(USB-Blaster[™] もしくは USB-Blaster[™] II)を接続し、Quartus[®] Prime 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 ⇒ Programmer を選択します。 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: <u>F</u> ile <u>E</u> dit <u>V</u> iew	:/Lab/nios2_basic_prj/nio P <u>r</u> ocessing <u>T</u> ools <u>W</u> ine	s2_basic_lab - nios dow <u>H</u> elp	s2_basic_lab -	[Chain2.cdf]*					Search a	altera.cor	n S
Hardware Setup	USB-Blaster [USB-0]	gramming when av	ailable	Mode: 🗍	TAG		• 1	Progress:	100%	Success	ful)
Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine	Security Bit	Erase	ISP CLAMP
Stop	output_files/nios2_bas	5CGXFC5C6F27	037C8AA5	037C8AA5	V						
Add Delete Add File Add File Add File Add Device Add Device I Down		F27									
											t



3. <u>ソフトウェア</u>

- 3-1. Nios®IISBT 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 オルダを指定します。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.	
Workspace: C:¥Lab¥nios2_basic_prj¥software	✓ <u>B</u> rowse
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。

	n rempiate	
ios II Software Examples		
Create a new application and bo	ard support package based on a software example template	
Target hardware information		
SOPC Information File name:	C:¥Lab¥nios2_basic_prj¥nios2_system.sopcinfo	
CPIL name:		
CPO name:	nios2_gen2_0	
Application project		
Project name: soft_test		
<u> </u>		
Vise default location		
Project location: C:¥Lab¥	nios2_basic_prj¥software¥soft_test	
Project template		
Templates	Template description	
Blank Project	Blank Project creates an empty project to which you 🔺	
Board Diagnostics	can add your code.	
Float2 Functionality	E	
Float2 GCC	to the readme.txt file in the project directory.	
Float2 Performance		
Hello Freestanding	The BSP for this template is based on the Altera HAL	
	operating system. To use a RSD based on a different	
Hello MicroC/OS-II	operating system. To use a DSP based on a different	
Hello MicroC/OS-II	operating system, to use a BSP base on a difference operating system, click Next and select the BSP from	
Hello MicroC/OS-II	operating system, click Next and select the BSP from	
Hello MicroC/OS-II	operating system, click Next and select the BSP from	
Hello MicroC/OS-II	operating system, click Next and select the BSP from	
Hello MicroC/OS-II	operating system, click Next and select the BSP from	
Hello MicroC/OS-II	operating system, click Next and select the BSP from	

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 ウィンドウの、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	- • ×
Select a wizard Create a new source file	
<u>W</u> izards:	
type filter text	
 Nios II Application Nios II Application and BSP from Template Nios II Library General C/C++ C Project C++ Project C Croyert to a C/C++ Autotools Project Convert to a C/C++ Project (Adds C/C++ Nature) File from Template Folder Header File Makefile Brotert with Existing Code Source File 	E
(?) < Back	Cancel



New Source	: File				
Source File Create a nev	w source file.				
Source folder:	soft_test				Browse
Source file:	soft_test.c				
<u>T</u> emplate:	Default C++ source te	mplate			▼ Configure
-					
?		< <u>B</u> ack	Next >	Einish	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。

```
#include <stdio.h>
#include <unistd.h>
#include "system.h"
#include "altera_avalon_pio_regs.h"
int main()
{
    printf( "Hello from Nios II! Start!\n");
    while(1)
    {
        IOWR_ALTERA_AVALON_PIO_DATA(PIO_0_BASE, 0x55);
        usleep(500000);
        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。

iun portruici denuges port	vers Linker Script Enable File Ger	neration Target BSP Directory			
inker Section Mappings					
Linker Section Name	Linker Re	gion Name	Memory Device Name		Add
.bss	onchip m	emory2 0	onchip memory2 0		Remove
entry	reset		onchip_memory2_0		Restore Defaults
exceptions	onchip_m	emory2_0	onchip_memory2_0		
.heap	onchip_m	emory2_0	onchip_memory2_0		
.rodata	onchip_m	emory2_0	onchip_memory2_0		
.rwdata	onchip_m	emory2_0	onchip_memory2_0		
.stack	onch i p_m	emory2_0	onchip_memory2_0		
.text	OUCUID W	emory2 U	Onchip memory2 0		
Linker Memory Regions				H. C.	
Linker Region Name	Address Range	Memory Device Name	Size (bytes)	Offset (bytes)	Add
onchip memory2 0	0x00020020 - 0x0003F	FFF onchip memory2 0	131040		32 Remove
reset	0×00020000 - 0×00020	11F onchip memory2 0	32		0 Restore Defaults
					Add Memory Device
					Add Memory Device Remove Memory Device Memory Usage
					Add Memory Device Remove Memory Device Memory Usage
					Add Memory Device Remove Memory Device Memory Usage Memory Map
raved out entries are auton	natically created at generate tim	e. They are not editable or persiste	ed in the BSP settings file.		Add Memory Device Remove Memory Device Memory Usage Memory Map
Srayed out entries are auton	natically created at generate tim	ie. They are not editable or persiste	ed in the BSP settings file.		Add Memory Device Remove Memory Device Memory Usage Memory Map
rayed out entries are auton	natically created at generate tim	ie. They are not editable or persiste	ed in the BSP settings file.		Add Memory Device Remove Memory Device Memory Usage Memory Map
Srayed out entries are auton nformation Problems Proces Setting Tail Inker.interrupt	natically created at generate tim singregion_name" set to	e. They are not editable or persiste	ed in the BSP settings file.		Add Memory Device Remove Memory Device Memory Usage Memory Map
Frayed out entries are auton nformation Problems Proces Setting Thal.Inker.exception Setting Thal.Inker.exception	natically created at generate tim sing stack_memory_region_name*set tr _stack_memory_region_name*set i	e. They are not editable or persiste o "onchip_memory2_0". to "onchip_memory2_0".	ed in the BSP settings file.		Add Memory Device Remove Memory Device Memory Usage Memory Map
rayed out entries are auton nformation <u>Problems</u> Proces Setting Thal.Inker.interrupt_ Setting Thal.Inker.exception Loading drivers from ensemb	natically created at generate tim ssing stack_memory_region_name" set to stack_memory_region_name" set i ele report.	ie. They are not editable or persiste o "onchip_memory2_0". to "onchip_memory2_0".	ed in the BSP settings file.		Add Memory Device Remove Memory Device Memory Usage Memory Map
irayed out entries are auton nformation Proces Setting "hal.inker.interrupt_ Setting "hal.inker.exception Loading drivers from ensem Mapped module: "hios2_gen	natically created at generate tim sing stack_memory_region_name" set to _stack_memory_region_name" set se report. 2_0" to use the default driver versio	ie. They are not editable or persiste o "onchip_memory2_0". to "onchip_memory2_0". on.	ed in the BSP settings file.		Add Memory Device Remove Memory Device Memory Usage Memory Map
rayed out entries are auton formation Proces Setting 'hal.linker.interrupt_ Setting 'hal.linker.exception Loading drivers from ensemb Mapped module: 'hios2_gen Mapped module: 'hios2.gen	natically created at generate tim sing stack_memory_region_name" set to _stack_memory_region_name" set to le report. 2_0" to use the default driver version.	e. They are not editable or persiste "onchip_memory2_0". to "onchip_memory2_0". on.	ed in the BSP settings file.		Add Memory Device Remove Memory Device Memory Usage Memory Map
irayed out entries are auton formation Problems Proces Setting Thal.Inker.exception Setting drivers from ensemb Mapped module: Tag.uart. Mapped module: Tag.uart.	natically created at generate tim sing stack_memory_region_name" set to _stack_memory_region_name" set le report. 2_0" to use the default driver version. 0" to use the default driver version.	e. They are not editable or persiste o "onchip_memory2_0". to "onchip_memory2_0". on.	ed in the BSP settings file.		Add Memory Device Remove Memory Device Memory Usage Memory Map
Srayed out entries are auton nformation <u>Problems</u> Proces Setting "hal.linker.interrupt_ Setting "hal.linker.exception Loading drivers from ensemb Mapped module: "pio_0" to L Mapped mo	natically created at generate tim ssing stack_memory_region_name" set to stack_memory_region_name" set le report. 2_0" to use the default driver version. o" to use the default driver version n ensemble report.	ie. They are not editable or persiste o "onchip_memory2_0". to "onchip_memory2_0". on.	ed in the BSP settings file.		Add Memory Device Remove Memory Device Memory Usage Memory Map
Srayed out entries are auton nformation Problems Proces 9 Setting 'hal.linker.interrupt_ 9 Setting 'hal.linker.exception 1 Loading drivers from ensemt 9 Mapped module: 'hio22_gent 9 Mapped module: 'hio24_uent 9 Finished loading drivers from see 9 Loading BSP settings from see	natically created at generate tim sing stack_memory_region_name" set to _stack_memory_region_name" set is le report. 2.0° to use the default driver version. 0° to use the default driver version. 0° to use the default driver version. ensemble report. ettings file.	e. They are not editable or persiste "onchip_memory2_0". to "onchip_memory2_0". on.	ed in the BSP settings file.		Add Memory Device Remove Memory Device Memory Usage Memory Map

⑨ ソフトウェアをビルドします。Nios®IISBT 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。



ALTIMA

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 オルダ(_bsp がついていない方)をハイライトし、右クリック ⇒ Run As ⇒ Nios II Hardware をクリックします。

	Build Configurations Make Targets Index	+ + +			
	Show in Remote Systems view Profiling Tools Convert To Profile As	•] Ca	onsole 🕸 🔲 Properties	
	Debug As	►	est	2]	
	Run As	►	4	1 Lauterbach ISS	init
	Compare With	•	C	2 Local C/C++ Application	
	Restore from Local History			3 Nios II Hardware	our
	Nios II	→	1	4 Nios II ModelSim	
*	Update Linked Resources Run C/C++ Code Analysis			Run Configurations	

② Nios®II Console ウィンドウに、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。



上図のように「Hello from Nios II! Start!」とキャラクタが出力(表示)されたら、③へ進みます。

上図のようなキャラクタが出力(表示)されなくて Run Configurations が立ち上がった場合、Target Connection タ ブを選択して、右上の Refresh Connections ボタンをクリックします。

Run Configurations							
Create, manage, and run configurations ONO Nios II target connection paths were	s re located. Check connections	and that a Nios II .sof	is downloaded.				
Image: Second secon	Name: soft_test Nios II Proje t La Target Cor Connections Processors:	Hardware configuration	Davies ID	Source	Name	Arabilantura	Refresh Connections
Nios II Hardware	Cable	Device	Device iD	instance ib	Name	Architecture	Resolve Names
📩 soft_test Nios II Hardware con	fiel L						System ID Properties
📩 Nios II Hardware v2 (beta)	Byte Stream Devices:						
Nios II ModelSim	Cable	Device	Device ID	Instance ID	Name	Version	
	Disable 'Nios II Console' Uuartus Project File name < System ID checks Ignore mismatched syst Download Download Output Download ELF to select Reset the selected target	view Using default.sopcinfo&.jdi em ID em timestamp ed larget system t system	files extracted fror	1 ELF >			
Filter matched 9 of 9 items	5					Apply	Re <u>v</u> ert
?						Run	Close

下図のようにインテル FPGA ダウンロード・ケーブル(USB-Blaster[™] など)を検出してターゲット・ボードとの接続 が確認できたら、Ignore mismatched system ID と Ignore mismatched system timestamp の両方にチェックを入れて、 右下にある Apply ボタンをクリック後に、その下の Run ボタンもクリックします。

Run Configurations		
Create, manage, and run configurations Connected system ID hash not found on the	arget at expected base address.	
	Name: soft_test Nios II Hardware configuration Project Surget Connection Project Surget Connection Processors: Cable Device	Refresh Connections
Nos II Hardware Nos II Hardware confi Nos II Hardware v2 (beta) Nos II ModelSim Nos II ModelSim Nos II ModelSim v2 (beta)	CDI-Directer on Jocs Hoot, (CDI-D) SCCTPEX(CS)1 I nice2 ren3 1 Nice2 Byte Stream Devices: Cable Device D Vestance D Name Cable Device D Vestance D Name Vestance D Cable Device D Vestance D Name Vestance D Cable Device D Vestance D Name Vestance D Cable Device D Vestance D Iterative D Vestance D Cable Device D Vestance D Iterative D Vestance D Cable Device D SCCTPEX(CS)1 I Iterative D Disable Nos I Consold View Vestance D Vestance D Vestance D	Resolve Names System D Properties.
	Guartus Project File name; < Using default .opcinto & .gli ties extracted from ELF > System ID checks	
Filter matched 9 of 9 items	Apply	Regert
0	Bun	Close

この資料の説明では、Qsys システムに System ID Peripheral を追加していませんが、事前に追加しておくとこの 接続できないエラーが回避できます。

System ID Peripheral - sysid_qsys_0	×
System ID Peripheral altera_avalon_sysid_qsys	Documentation
Block Diagram Show signals Show signals clk clk reset control_slave avaion attera_avalon_sysid_qsys	Parameters 32 bit System ID: 0x0000000 Description Please use hexadecimal numbers only in System ID.
 Info: sysid_qsys_0: System ID is not assigned aut Info: sysid_qsys_0: Time stamp will be automatical 	omatically. Edit the System ID parameter to provide a unique ID ally updated when this component is generated. Cancel Finish

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 Make Targets Index	+ + +			
Show in Remote Systems view Profiling Tools Convert To Profile As	+	Cor	nsole 🖾 🔲 Properties	
Debug As	•	C	1 Local C/C++ Application	
Run As	×		2 Nios II Hardware	it
Compare With Restore from Local History	•	blet	Debug Configurations	ur

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

Confirm	n Perspective Switch	3
?	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?	
Reme	ember my decision	
	<u>Y</u> es <u>N</u> o	

⑤ Nios®IISBT のウィンドウがデバッグ・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ィ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。



⑥ デバッガのアイコンの情報は、以下の通りです。



🛆 ALTIMA



<u> 改版履歴</u>

Revision	年月	概要
1	2017 年 11 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。
 株式会社マクニカ アルティマ カンパニー https://www.alt.macnica.co.jp/ 技術情報サイト アルティマ技術データベース http://www.alt.macnica.co.jp/ 技術情報サイト アルティマ技術データベース http://www.alt.macnica.co.jp/ 技術情報サイト ETS https://www.alt.macnica.co.jp/ 技術情報サイト ETS https://www.alt.macnica.co.jp/ https://www.alt.macnica.co.jp/ https://www.alt.macnica.co.jp/ https://www.alt.macnica.co.jp/ https://www.alt.macnica.co.jp/ https://www.alt.macnica.co.jp/

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。