

# Quartus Prime ガイド Qsys システム統合ツールの使い方

Ver.16



ELSENA

ELSENA,Inc.





# Quartus Prime ガイド

# Qsys システム統合ツールの使い方

<u>日次</u>	
1. はじめに	4
2. Qsys の基本操作	6
2-1. Qsys システムの新規作成	6
2-2. 既存の Qsys システムを開く	6
2-3. Qsys <b>システムの</b> 保存	7
2-4. Qsys の終了	7
3. コンポーネントの追加	8
3-1. <b>コンポーネントの</b> 検索	
3-2. コンポーネントの追加と各種設定	
3-3. コンポーネントの並べ替え	
4. コンポーネントの接続と設定	11
4-1. クロックとリセット	11
4-1-1. 周波数の設定	11
4-1-2. 入力ポート	
4-1-3. 出力ポート	
4-1-4. 各コンポーネント間のクロック接続	
4-2. マスタとスレーブ間の接続	17
4-3. 割り込み	
4-3-1. 割り込みの接続	
4-3-2. 割り込みの設定	
4-4. Qsys システムの外部に出すピン	
5. Qsys システムの設定と生成	20
5-1. ベース・アドレスの設定	
5-2. Qsys インタコネクトのオプション設定	
5-3. Qsys システムの生成	
5-4. Qsys システムのテストベンチ・システム生成	
5-5. Qsys システムのインスタンスを含んだテンプレート	





# Quartus Prime ガイド Qsys システム統合ツールの使い方

6. Quartus Prime での各種設定	26
6-1. Qsys システム用ファイルの登録	26
6-2sdc ファイルの登録	26
改版履歴	27



## 1. <u>はじめに</u>

この資料は、Quartus<sup>®</sup> Prime 開発ソフトウェア(以下、Quartus Prime)を使用するユーザ向けの資料です。FPGA / CPLD の開発フローの中で、主に下図の赤枠内の開発フェーズで非常に参考になる資料です。



Qsys システム統合ツール(以下、Qsys)は、各コンポーネント(Intellectual Property (IP) ファンクションやサブシ ステム、ユーザ・ロジックなど)を接続するインタコネクト・ロジック(アドレス/データ・バスの接続やバス幅整合ロ ジック、アドレス・デコーダ・ロジック、アービトレーション・ロジックなど)を自動的に生成して、FPGA デザイン・プロ セスの時間と労力を大幅に削減します。Qsys は FPGA に最適化されたネットワーク・オン・チップ(NoC)の新しい 技術で開発されており 前世代ツールの SOPC Builder と比べて、性能の向上やデザインの再利用のしやすさ、 そして検証の迅速化を実現しています。



Qsys システム内にはマスタとスレーブが存在し、各コンポーネント間のインタフェース(Qsys インタコネクト)は、 Avalon<sup>®</sup> や ARM<sup>®</sup> AMBA<sup>®</sup> AXI™、AMBA APB™、AMBA AHB™ などの各種業界標準インタフェースの混在 をサポートします。



また、Qsys はソースとシンクのようなストリーミング接続にも対応しています。



Nios<sup>®</sup> II プロセッサ(以下、Nios II)や FPGA デバイスに内蔵された ARM<sup>®</sup> コア内蔵の SoC FPGA を使用す る場合などに限らず、最近では Qsys を使用するケースが多くあるため、この資料では Qsys の簡単な使い方を 説明します。

なお、各コンポーネントの詳細については、各プロセッサや各ペリフェラルのドキュメントを参照してください。



### 2. <u>Qsys</u> の基本操作

Qsys は Quartus Prime から起動します。Quartus Prime の Tools メニュー  $\Rightarrow$  Qsys を選択するか  $\stackrel{\blacktriangle}{=}$  ツー ルバーをクリックしてください。

Qsys - unsaved.qsys* (E:¥Altera¥unsa File Edit System Congrate View Tool	aved.q	sys)							
		System	Contents	Address Man	Interconnect Requirements				-*
			Sy	/stem: unsaved	and connect requirements				
Project	+	Use	Conn	Name	Description	Export	Clock	Base	
Library	$\times$	<b>V</b>	⊳	⊟ clk_0 clk_in	Clock Source Clock Input	clk	exported		
Basic Functions     DSP     Interface Protocols			×	clk_in_reset clk	Reset Input Clock Output	reset Double-click to export	clk_0		
Low Power     Memory Interfaces and Controller:				clk_reset	Reset Output	Double-click to export			
< >	Ŧ								
New Edit + Add									
iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii									
enseved [anseved [asseved [asseved]] enseved [anseved [asseved [asseved]] enseved [anseved [asseved [asseved]] enseved [anseved [asseved]] enseved [anseved [asseved]] enseved [asseved [asseved]] enseved [asseved [asseved]] enseved [asseve									
. <b>to</b> ck_0		۲ ا		Comment filters	III				4
	×= 1	Meccar		ourrent mer.					-*-
	Тура	903306	Path	Message					
0 Errors, 0 Warnings								Generate HDL	Finish

#### 2-1. Qsys システムの新規作成

上図が Qsys を新規に起動した状態です。

#### 2-2. 既存の Qsys システムを開く

既存の Qsys システムを開くには、Qsys の File メニュー  $\Rightarrow$  Open を選択して、該当の .qsys ファイルを選択 してください。



#### 2-3. Qsys システムの保存

Qsys システムを保存するには、Qsys の File メニュー ⇒ Save(上書き保存)、または Save as(名前を付けて 保存)を選択してください。下のようなウィンドウが表示されて Close がクリックできるようになったらクリックしま す。



#### 2-4. Qsys の終了

Qsys を終了する時に、設定や接続を変更してから Qsys システムの生成を行わないまま終了しようとすると、 下のようなメッセージが表示されます。Qsys システムの生成を行ってから終了するか、もしくは生成しないで終了 するかを決めてください。(Qsys システムの生成については、後述の「5. Qsys システムの設定と生成」で説明して います。)

ĺ	Generat	te Now?
	?	Recent changes have not been generated. Generate now?
		(はい(Y) いいえ(N) 取消

## 3. <u>コンポーネントの追加</u>

Quartus Prime(Qsys)に予め用意されているコンポーネントから、使用するコンポーネントを検索してシステムに 追加していきます。コンポーネントの検索や追加、各種設定などについて説明します。

※ 自作のコンポーネントを追加登録して Qsys システム内で使用することもできますが、この Component Editor についてはこの資料では触れません。

#### 3-1. コンポーネントの検索

IP Catalog ウィンドウにカテゴリ毎に分類されているのでその中から探すか、または検索バーにコンポーネント 名の一部でも入力すると該当するコンポーネントがフィルタリングされて表示されるので、該当のコンポーネントを ハイライトして Add をクリック、もしくはダブルクリックします。



#### 3-2. コンポーネントの追加と各種設定

Add すると、追加しようとしているコンポーネントの設定画面が開きます。まずは、設定できる箇所だけ設定して ください。エラーが残っている状態でも、とりあえず Qsys システム内に追加することができます。(この後に説明 する接続などができていないと、完全にエラーが消えない場合があります。)

※ ここでは、各コンポーネントの設定画面の詳細については説明しません。設定の詳細については、各コン ポーネントのドキュメントを参照してください。 必要な設定ができたら、Finish をクリックします。

Nios II Processor - nios2_gen2_0			×
Nios II Processor altera_nios2_gen2		例)Nios II プロセッサ	Documentation
Block Diagram         Bhock Diagram <td< th=""><th>Main         Vectors         Caches and Memory Interfaces         Arithmetic Instructions                • Reset Vector Reset vector memory: Reset vector offset: • 0x0000000 Reset vector: • 0x0000000 • Exception Vector Exception vector offset: • 0x00000000          None • 0x0000000 • Caches and Memory: • 0x00000000                 • Fast TLB Miss Exception Vector Fast TLB Miss Exception vector memory: • 0x00000000          None • Caches and Memory Interfaces • 0x0000000                 • Fast TLB Miss Exception vector memory: • Fast TLB Miss Exception vector: • 0x00000000                 • Fast TLB Miss Exception vector: • 0x00000000</th><th>MMU and MPU Settings   JTAG Debug   Advanced Features</th><th></th></td<>	Main         Vectors         Caches and Memory Interfaces         Arithmetic Instructions                • Reset Vector Reset vector memory: Reset vector offset: • 0x0000000 Reset vector: • 0x0000000 • Exception Vector Exception vector offset: • 0x00000000          None • 0x0000000 • Caches and Memory: • 0x00000000                 • Fast TLB Miss Exception Vector Fast TLB Miss Exception vector memory: • 0x00000000          None • Caches and Memory Interfaces • 0x0000000                 • Fast TLB Miss Exception vector memory: • Fast TLB Miss Exception vector: • 0x00000000                 • Fast TLB Miss Exception vector: • 0x00000000	MMU and MPU Settings   JTAG Debug   Advanced Features	
			Cancel Finish

#### すると、Qsys システム内にコンポーネントが追加されます。

& Qsys - unsaved.qsys* (D:¥work¥c5_qsys_1600¥unsaved.qsys) Elle Edit System Generate View Iools Help					例)Nios II プロセッサ			
📑 IP Catalog 😣 🛛 🔤 📑	🛄 Sys	tem Contents 🙁 Addre	ss Map 💠 Interconnect	Requirements 🛛	L			
🔍 nios ii 🛛 🗙 🔯	I	🔺 😻 System: unsave	d Path:nios2_gen2_0					
Altera Nios II Custom Instruction	+ U	e Connections	Name	Description	Export	Clock	Base End	IRQ T
Altera Nios II Custom Instruction :	<b>1</b>	2	🗆 clk_0	Clock Source				
Processors and Peripherals	×	C	clk_in	Clock Input	clk	exported		
- Vo-Processors	2	°_−□	clk_in_reset	Reset Input	reset			
Bitswap	<b>T</b>		clk	Clock Output	Double-click to expo	rt clk_0		
Custom Instruction Interconnect		7	E III pige 2 gen 2 fl	Nice TI Processor				
Custom Instruction Master Transla			clk	Clock Input	Double-click to expo	t unconnecte		
Eloating Point Hardware			reset	Reset Input	Double-click to expo	t [ck]	追加されたコン	パーネント 📗
Floating Point Hardware 2			data_master	Avalon Memory Mapped Master	Double-click to expo	rt [clk]		
Embedded Processors			instruction_master	Avaion Memory Mapped Master	Double-click to expo	rt [clk]		
Nios II (Classic) Processor		×	irq	Interrupt Receiver	Double-click to expo	nt [clk]	IRQ 0	IRQ 31
NIOS I Processor			debug_reset_request	Reset Output	Double-click to expo	rt [clk]		
		••	debug_mem_slave	Avalon Memory Mapped Slave	Double-click to expo	rt [clk]		
New, Edit. + Add.		×—	custom_instruction_m	Custom Instruction Master	Double-click to expo	rt		
				III				F I
🧏 Hierarchy 🛛 Device Family 🌣 🗕 🗗 🗖	n,	非·ママ 🐨 Current fill	er:					
🖳 unsaved [unsaved.qsys*]	<mark>ŏ</mark> ≣ Me	sages 🛛						- 5 0
🕮 🍽 reset	Туре	Path	Message					
	- 🔀	4 Errors						
E Connections	8	unsaved_nios2_gen	2_0 Reset slave is not spec	ified. Please select the reset slave				
	8	unsaved_nios2_gen	2_0 Exception slave is not s	pecified. Please select the exception sl	ave			
	8	unsaved nios2 gen	2 0 nios2 gen2 0.clk mus	t be connected to a clock output				
	8	unsaved_nios2_gen	2_0 nios2_gen2_0.reset m	ust be connected to a reset source				
4 Errors, 0 Warnings								Generate HDL Finish

追加したコンポーネントの名前をデフォルトから変更したい場合は、該当のコンポーネントをハイライトして、右ク リック ⇒ Rename を選択して、名前を修正してください。



この手順を繰り返して、使用したいコンポーネントをすべて Qsys システム内に追加してください。





追加したコンポーネントの順序を並べ替えたい時は、該当のコンポーネントをハイライトした状態で 🏊 や 📤、 💟 💟 などで順序を変更することができます。



# 4. コンポーネントの接続と設定

使用するコンポーネントの追加が終わったら、コンポーネント間の接続を行います。クロック同士やリセット同士、 マスタとスレーブ、ソースとシンクなど同じ種類のインタフェースの接続を行います。また、ストリーミング接続にも 対応しています。

- ※ Qsys システムに複数のクロックが供給される場合は、複数個の Clock Source を Qsys システムに追加し て、仕様に応じて必要な設定や接続を行ってください。この資料では、1 種類のクロック(50MHz)が Qsys システムへ供給され、更に PLL で異なる 2 種類の周波数を生成してから各コンポーネントへ供給すると いった仕様のデザインを例にしています。
- ※ Nios II プロセッサの Reset Vector や Exception Vector などは、Qsys システム上の接続を行わないと設定 ができなかったり、エラーが消えなかったりします。これは一旦、エラーが残っている状態で Qsys システ ムへの Nios II プロセッサの追加を済ませた後に Qsys システム内の接続を適切に行うことで、設定がで きるようになり、エラーが消えます。この資料では詳細な説明を省略していますので、詳しくは各コンポーネ ントのドキュメントを参照してください。

#### 4-1. クロックとリセット

クロックとリセットの設定や接続を行います。

#### 4-1-1. 周波数の設定

Qsys システムへ供給されるクロックを設定します。Clock Source をダブルクリック または 右クリック ⇒ Edit を選択します。各種設定が終わって設定画面を閉じる場合は、Parameters タブの × をクリックします。

13	System	Contents 🛛 🕄	Address Map 🛛	Interconnect Requirements	: 22	- d 🗆	🦄 Parameters			- d 🗆
		🛛 😻 System:	nios2_system Path	h:clk_0			System: nios2_sys	stem Path: c	閉じるなら ×	
+	Use	Connections		Name	Description	Export	Clock Souro	e 📘		Details
	<b>V</b>	0.0		🗆 clk_0	Clock Source	<u>^</u>	Parameters			
			· · · ·	cik_in cik_in_reset	ock input	reset	Clock frequency	× 500000	100 Hz	
				clk		Double-click to e:	Clock frequ	iency is knowni		
				ck_reset Ø	フルクリック	Double-click to e.	Reset synchroni	ous edges: None	▼	
-				refclk		Double-click to e:				-
-			$\rightarrow$	reset	Reset Input	Double-click to e:				
				outclk0	Clock Output	Double-click to e:				
			ę	locked	Conduit	Double-click to e. Double-click to e.				

#### Clock frequency

この Clock Source へ入力されるクロックの周波数を設定します。単位は Hz なので、桁数に注意してください。

※ k を入力すると数秒後に 0 が 3 つ、m を入力すると数秒後に 0 が 6 個自動で追加されます。

Clock frequency is known

周波数がわかっている場合は、チェックを入れます。チェックをしない場合、Qsys システムの生成(Generate)に 失敗することがあるので、チェックを入れることを推奨します。

Reset synchronous edge

このクロック・ソースが期待するリセットの種類を選択します。

None	:リセットのアサートとディアサートがクロックに同期しない。(押しボタンによるリセットなど)
Both	:リセットのアサートとディアサートがクロックに同期する。
Deassert	:リセットのアサートは同期せず、ディアサートは同期する。

#### 4-1-2. **入力ポート**

Clock Source の入力ポートは Qsys システムへのクロック入力(Clock Input)とリセット入力(Reset Input)がデフ オルトで Export に設定済みです。(Export 欄が 太字 になっていると Exported、つまり Qsys システムの外部と 接続できるピンとして設定されていることを示しています。)

Use	Connections	Name	Description	Export	Clock
<b>V</b>		🗆 clk_0	Clock Source		
	°-°	clk_in	Clock Input	clk	exported
	· · ·	clk_in_reset	Reset Input	reset	
		clk	Clock Output	Double-click to export	clk_0
		clk_reset	Reset Output	Double-click to export	

#### 4-1-3. 出**力ポート**

Clock Source の出力ポートは Clock Source を介して Qsys システム内の他のコンポーネントへ供給するクロック出力(Clock Output)とリセット出力(Reset Output)があります。

Use	Connections	Name	Description	Export	Clock
<b>V</b>		🗆 clk_0	Clock Source		
		clk_in	Clock Input	clk	exported
	· · · · · · · · · · · · · · · · · · ·	clk_in_reset	Reset Input	reset	
		clk	Clock Output	Double-click to export	clk_0
		clk_reset	Reset Output	Double-click to export	

#### 4-1-4. 各コンポーネント間のクロック接続

Clock Source を介して Qsys システム内の他のコンポーネントへ供給するクロックを、供給先のコンポーネント のクロック入力と接続します。

この例では FPGA 内部の PLL で生成した異なる周波数のクロックを Qsys システム内の各コンポーネントへ 供給しているので、まずは Clock Source のクロック出力と PLL のクロック入力を接続します。(PLL を使用しな い Qsys システムの場合は、一般的に Clock Source のクロック出力と他の各コンポーネントの入力クロックを接続します。)

Use	Connections	Name	Description	Export	Clock
<b>V</b>		🗆 clk_0	Clock Source		
	□ 未接続□-	clk_in	Clock Input	clk	exported
		clk_in_reset	Reset Input	reset	
		clk	Clock Output	Double-click to export	clk_0
		clk_reset	Reset Output	Double-click to export	
<b>V</b>		🗆 pll	Altera PLL		
(		refolk	Clock Input	Double-click to export	unconnected
		reset	Reset Input	Double-click to export	
		outclk0	Clock Output	Double-click to export	pll_outclk0
		outclk 1	Clock Output	Double-click to export	pll_outclk1
	<b></b> ~	locked	Conduit	Double-click to export	
		_			



Use	Connections	Name	Description	Export	Clock
<b>V</b>		🗆 clk_0	Clock Source		
	│ <mark>──接続</mark> ──────────────────────────	clk_in	Clock Input	clk	exported
		clk_in_reset	Reset Input	reset	
		clk	Clock Output	Double-click to export	clk_0
		clk_reset	Reset Output	Double-click to export	
1		🖯 pll	Altera PLL		
(		refolk	Clock Input	Double-click to export	clk_0
	$\uparrow \diamond \rightarrow \rightarrow$	reset	Reset Input	Double-click to export	
		outclk0	Clock Output	Double-click to export	pll_outclk0
		outclk 1	Clock Output	Double-click to export	pll_outclk1
		locked	Conduit	Double-click to export	

次に、PLL で生成した 2 種類のクロックをシステムの仕様に応じて各コンポーネントのクロック入力へ接続します。ここで、表示のフィルタ機能を使用してクロックとリセット関係のみの表示(Clock and Reset Interfaces)にすると、 接続がやりやすくなります。

Use	Connections	Name	Description	Export	Clock
V		⊟ clk_0	Clock Source		
	<u> </u>	clk_in	Clock Input	cik	exported
	° 🗗	clk_in_reset	Reset Input	reset	
		clk	Clock Output	表示のフィルタ・Clock and Rese	t Interface
		clk_reset	Reset Output		tilleridee
1		🖯 pll	Altera PLL		
	$  \bullet + \diamond \diamond + \rightarrow$	refclk	Clock Input	Double-click to expe	rt clk_0
	$   \diamond + + \diamond \rightarrow$	reset	Reset Input	Double-click to expe	rt
	>++	outclk0	Clock Output	Double-click to expe	rt pll_outelk0
		outclk1	Clock Output	Double-click to expe	rt pll_outelk1
<b>V</b>		日 🖳 nios2_cpu	Nios II Processor		
	⊹ <b>(+)</b> →	clk	Clock Input	Double-click to expe	rt pll_outclk0
	││ <b>०<del>─</del>─</b> + <i></i> ↔→	reset	Reset Input	Double-click to expe	rt [clk]
	$  $ $ $ $ $ $\vdash$	debug_reset_request	Reset Output	Double-click to expe	rt [clk]
1		🗆 jtag_uart	JTAG UART		
		clk	Clock Input	Double-click to expe	rt pll_outclk0
	♦┭+♦→	reset	Reset Input	Double-click to expe	rt [clk]
<b>V</b>		onchip_memory	On-Chip Memory (RAM o	r ROM)	
	$ $ $\leftrightarrow$ $\rightarrow$	clk1	Clock Input	Double-click to expe	rt pll_outclk0
	$   \diamond + + \diamond \rightarrow$	reset1	Reset Input	Double-click to expe	rt [clk1]
1		日 喧 sram	Generic Tri-State Control	ller	
		clk	Clock Input	Double-click to expe	rt pll_outclk0
	$   \diamond + \diamond \rightarrow$	reset	Reset Input	Double-click to expe	rt [clk]
<b>V</b>		🗆 ext_sram_bus	Tri-State Conduit Bridge		
	$ $ $\leftrightarrow$ $\rightarrow$	clk	Clock Input	Double-click to expe	rt pll_outclk0
	$   \diamond + \diamond \rightarrow$	reset	Reset Input	Double-click to expe	rt [clk]
<b>V</b>		曰 唱 epcq	Altera Serial Flash Contro	oller	
	$  \leftrightarrow ( \bullet ) \rightarrow$	clock_sink	Clock Input	Double-click to expe	rt pll_outclk1
	$   \diamond + + \diamond \rightarrow$	reset	Reset Input	Double-click to expe	rt [clock_sink]
<b>V</b>		🗆 led_pio	PIO (Parallel I/O)		
	$\diamond \leftrightarrow \rightarrow$	clk	Clock Input	Double-click to expe	rt pll_outclk0
	$\rightarrow \rightarrow \rightarrow$	reset	Reset Input	Double-click to expe	rt [clk]

#### 【補足①:表示のフィルタ機能】

表示のフィルタ機能を使用して表示を絞り込むことで、接続などの時に見やすくなります。



🔓 Filters		×
Filter: All Interfaces All Interfaces Clock and Reset In Avalon-MM Interfac Match Hide Resets What Hide Clocks Hide Clocks and Re not	rfaces ss s Value sets	Remove
Add Remove		
and match all:	uu Value	
no rules		
Add Remove		

# 

#### 【補足②:接続の方法】

先ほど、Connections 欄で O をクリックして接続する方法を紹介しました。この方法以外に、何通りか接続の仕方がありますので、紹介します。お好みや必要に応じて、使い分けてください。

◆ <u>クロック入力の Clock 欄で選択する方法</u>

クロックの供給を受ける側の各コンポーネントのクロック入力の Clock 欄をクリックすると、接続できる候補が 表示されるので、その中から選びます。

Use	Connections	Name	Description	Export	Clock
<b>V</b>		🖂 clk_0	Clock Source		
	P	clk_in	Clock Input	clk	exported
	· · · ·	clk_in_reset	Reset Input	reset	
		clk	Clock Output	Double-click to export	clk_0
		clk_reset	Reset Output	Double-click to export	
1		🖯 pll	Altera PLL		
	$  \bullet   \circ \circ \longrightarrow$	refclk	Clock Input	Double-click to export	clk_0
	$   \diamond \rightarrow \rightarrow$	reset	Reset Input	Double-click to export	
		outclk0	Clock Output	Double-click to export	pll_outclk0
		outclk1	Clock Output	Double-click to export	pll_outclk1
	· · · · · · · · · · · · · · · · · · ·	locked	Conduit	Double-click to export	
<b>V</b>		日咀 nios2_cpu	Nios II Processor		
		ok	Clock Input	Double-click to export	pll outcl 👻
	$    \diamond   \rightarrow \diamond \rightarrow \rightarrow \diamond \rightarrow \rightarrow \diamond \rightarrow \rightarrow \diamond \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow $	reset	Reset Input	Double-click to export	clk_0
		data_master	Avalon Memory Mapped Master	Double-click to export	pll_outclkU
		instruction_master	Avalon Memory Mapped Master	Double-click to export	
		irq	Interrupt Receiver	Double-click to export	[clk]
		debug_reset_request	Reset Output	Double-click to export	[clk]
		debug_mem_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]
		custom instruction m	Custom Instruction Master	Double-click to export	

◆ クロック出力の右クリック ⇒ Connections で選択する方法

クロックを供給する側の各コンポーネントのクロック出力をハイライトして 右クリック ⇒ Connections を選択す ると、接続できる候補が表示されるので、その中から選びます。



続いて、リセットの接続も同じように行います。

Use	Connections	Name	Description	Export	Clock
<b>V</b>		⊟ clk_0	Clock Source		
		clk_in	Clock Input	clk	exported
		clk_in_reset	Reset Input	reset	
		clk	Clock Output	Double-click to export	clk_0
		clk_reset	Reset Output		
<b>V</b>		🗆 pll	Altera PLL 衣示	のノイルタ: Clock and Reset In	nterface
	$  \bullet \rightarrow \circ \rightarrow \rightarrow$	refclk	Clock Input	Double-click to export	clk_0
		reset	Reset Input	Double-click to export	
	║┯≻┼┼┥	outclk0	Clock Output	Double-click to export	pll_outclk0
		outclk1	Clock Output	Double-click to export	pll_outclk1
<b>V</b>		🗆 🖳 nios2_cpu	Nios II Processor		
		clk	Clock Input	Double-click to export	pll_outclk0
		reset	Reset Input	Double-click to export	[clk]
		debug_reset_request	Reset Output	Double-click to export	[clk]
<b>V</b>		🗆 jtag_uart	JTAG UART		
		clk	Clock Input	Double-click to export	pll_outclk0
		reset	Reset Input	Double-click to export	[clk]
<b>V</b>		onchip_memory	On-Chip Memory (RAM or ROM)		
		clk1	Clock Input	Double-click to export	pll_outclk0
		reset1	Reset Input	Double-click to export	[clk1]
1		日 🖳 sram	Generic Tri-State Controller		
		clk	Clock Input	Double-click to export	pll_outclk0
		reset	Reset Input	Double-click to export	[clk]
<b>V</b>		🖯 ext_sram_bus	Tri-State Conduit Bridge		
		clk	Clock Input	Double-click to export	pll_outclk0
		reset	Reset Input	Double-click to export	[clk]
<b>V</b>		日 唱 epcq	Altera Serial Flash Controller		
		clock_sink	Clock Input	Double-click to export	pll_outclk1
		reset	Reset Input	Double-click to export	[clock_sink]
<b>V</b>		🗆 led_pio	PIO (Parallel I/O)		
	$  \rightarrow \rightarrow$	clk	Clock Input	Double-click to export	pll_outclk0
	$  \bigcirc \rightarrow$	reset	Reset Input	Double-click to export	[clk]

#### 4-2. マスタとスレーブ間の接続

マスタ側のコンポーネント(プロセッサなど)とスレーブ側のコンポーネント(ペリフェラルなど)の間のデータを接続します。複数のマスタや複数のスレーブの接続も可能です。

なお、使用するコンポーネントの仕様に合わせた接続が必要となります。そのため、この資料では詳細な説明を 割愛しています。(この Qsys システムの例では、下のように接続します。)

Use	Connections	Name	Description		Export	Clock
<b>V</b>		🖯 pll	Altera PLL			clk_0
	Ŷ	locked	Conduit			
1		曰 喧 nios2_cpu	Nios II Processor	表示の	ワイルタ: Hide Clock	and Reset
	$ \longrightarrow$	data_master	Avalon Memory Mapped	Master	Double-click to ex	port pll_outclk0
		instruction_master	Avalon Memory Mapped	Master	Double-click to ex	port [clk]
	$   \longrightarrow$	irq	Interrupt Receiver		Double-click to ex	port [clk]
	$  \bullet \bullet \longrightarrow$	debug_mem_slave	Avalon Memory Mapped	l Slave	Double-click to ex	port [clk]
	X	custom_instruction_m	Custom Instruction Mas	ter:	Double-click to ex	port
		🗆 jtag_uart	JTAG UART			[clk]
(	$\bullet$	avalon_jtag_slave	Avalon Memory Mapped	l Slave	Double-click to ex	port pll_outclk0
		irq	Interrupt Sender		Double-click to ex	port [clk]
<b>V</b>		onchip_memory	On-Chip Memory (RAM	or ROM)		[clk1]
(	$ \rightarrow $	s1	Avalon Memory Mapped	l Slave	Double-click to ex	port pll_outclk0
<b>V</b>		日望 sram	Generic Tri-State Contr	roller		[clk]
<b>(</b>	$[ \bullet \bullet ) \longrightarrow$	uas	Avalon Memory Mapped	l Slave	Double-click to ex	port pll_outclk0
		tom	Tristate Conduit Master	,	Double-click to ex	port [clk]
1		🗆 ext_sram_bus	Tri-State Conduit Bridg	e		[clk]
	$  \rightarrow \rightarrow$	tcs	Tristate Conduit Slave		Double-click to ex	port pll_outclk0
	↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓	out	Conduit		Double-click to ex	port
		曰 喧 epcq	Altera Serial Flash Con	troller		[clock_sink]
l (	$  \bullet \bullet \rangle$	avl_csr	Avalon Memory Mapped	l Slave	Double-click to ex	port pll_outclk1
	$\bullet \bullet \longrightarrow$	avl_mem	Avalon Memory Mapped	l Slave	Double-click to ex	port [clock_sink]
		interrupt_sender	Interrupt Sender		Double-click to ex	port [clock_sink]
		🗆 led_pio	PIO (Parallel I/O)			[clk]
(	$\bullet$	s1	Avalon Memory Mapped	l Slave	Double-click to ex	port pll_outclk0
		external_connection	Conduit		Double-click to ex	port



#### 4-3. 割り込み

Qsys システムによって、スレーブ側のコンポーネント(ペリフェラルなど)からマスタ側のコンポーネント(プロセッサなど)へ割り込みを通知するような構成の場合は、割り込み信号を接続することができます。

#### 4-3-1.割り込みの接続

IRQ 欄の O をクリックすると、割り込み信号が接続されます。

#### 4-3-2. 割り込みの設定

クリックした順に、0 から番号が自動で割り振られます。数字が小さいほど、優先順位の高い割り込みとなりま す。優先順位を変更したい場合は、数字をクリックして直接数字を変更してください。

Use	Connections	Name	Description	Export	Clock	Base	End	IRQ T
		🗆 pll	Altera PLL		clk_0			
	φ—	locked	Conduit	Double-click to export				
1		日 喧 nios2_cpu	Nios II Processor		[clk]			
		data_master	Avalon Memory Mapped Master	Double-click to export	pll_outclk0			
		instruction_master	Avalon Memory Mapped Master	Double-click to export	[clk]			
	$   \longrightarrow$	irq	Interrupt Receiver	Double-click to export	[clk]	IRQ	0 IRQ 3	
	<b>♦ ♦  </b> >	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				
V		🗆 jtag_uart	JTAG UART		[clk]			
	<b>♦                                   </b>	avalon_jtag_slave	Avalon Memory Mapped Slave	Double-click to export	pll_outclk0	■ 0x0000_0000	0×0000_0007	
		irq	Interrupt Sender	Double-click to export	[clk]			
<b>V</b>		onchip_memory	On-Chip Memory (RAM or ROM)		[clk 1]			
	<b>♦ ♦ Ⅰ</b> →	s1	Avalon Memory Mapped Slave	Double-click to export	pll_outclk0	₽° 0x0000_0000	0x0001_ffff	
		드 归 sram	Generic Tri-State Controller		[clk]			
	<b>♦ ♦ Ⅰ</b> →	uas	Avalon Memory Mapped Slave	Double-click to export	pll_outclk0	■ 0x0000_0000	0x000f_ffff	
		tom	Tristate Conduit Master	Double-click to export	[clk]			
<b>V</b>		🗉 ext_sram_bus	Tri-State Conduit Bridge		[clk]			
	$\diamond \rightarrow$	tcs	Tristate Conduit Slave	Double-click to export	pll_outclk0			
		out	Conduit	Double-click to export				
<b>V</b>		므 归 epcq	Altera Serial Flash Controller		[clock_sink]			
	$  \bullet \bullet \bullet \bullet \bullet \to$	avl_csr	Avalon Memory Mapped Slave	Double-click to export	pll_outclk1	■ 0x0000_0000	0×0000_001f	
	$  \bullet \bullet   \longrightarrow$	avl_mem	Avalon Memory Mapped Slave	Double-click to export	[clock_sink]	■ 0x0000_0000	0×01ff_fff	
	· ·	interrupt_sender	Interrupt Sender	Double-click to export	[clock_sink]			
<b>V</b>		🗆 led_pio	PIO (Parallel I/O)		[clk]			
	$ \bullet \circ \longrightarrow$	s1	Avalon Memory Mapped Slave	Double-click to export	pll_outclk0	■ 0x0000_0000	0×0000_000f	
	ф	external_connection	Conduit	Double-click to export				

#### 4-4. Qsys システムの外部に出すピン

FPGA 内のユーザ・ロジックと接続するために Qsys システムの外にピンを出したい場合は、Export 欄をダブ ルクリックするとピン名が編集できる状態になるので、ピン名を確定させます。



### 5. Qsys システムの設定と生成

コンポーネント間の接続が終わったら、Qsys システムの設定と生成(Generate)を行います。

#### 5-1. ベース・アドレスの設定

この状態でエラー・メッセージが残っている場合、ベース・アドレスが重複していることが多く見受けられます。重ならないように Base 欄を手動で編集しても良いですが、Qsys の System メニュー  $\Rightarrow$  Assign Base Addresses を選択すると、重複しないように自動で設定してくれます。

F	ile Edit	Syst	em Generate View Tools Help						
ſ	📩 IP Ca		Upgrade IP Cores		12	System	Conten	ts 🖾	Addr
			Assign Base Addresses	1				System:	nios2_
	Project		Assign Interrupt Numbers	ΓI	+	Use	Conne	ctions	
			Assign Custom Instruction Opcodes						
	€ Syst		Create Global Reset Network		×				<u>~</u>
	Library Bas DSP		Show System With Qsys Interconnect			<b>V</b>			
	ter ter ter ter		Remove Dangling Connections						$\rightarrow$
			Import Interface Requirements				++-	×	$\square$
		s Inte	rconnect		-				
	🗄 Univ	/ersity	/ Program				<b> </b>		$\rightarrow$
								┥───	

Address Map タブをクリックすると、設定されたベース・アドレスの一覧を確認することができます。

System Contents 🛛 💷 Addres	s Map 🖾 🛛 I	interconnect Requirements 🛛 🖾	
System: nios2_system Path: pll			
	nios2_cpu.dat	a_master	nios2_cpu.instruction_master
epcq.avl_csr	0×0424_1000	- 0×0424_101f	
epcq.avl_mem 0x0200		- 0×03ff_fff	0x0200_0000 - 0x03ff_fff
jtag_uart.avalon_jtag_slave	0×0424_1030	- 0×0424_1037	
led_pio.s1	0×0424_1020	- 0×0424_102f	
nios2_cpu.debug_mem_slave 0x0424		- 0×0424_0fff	0x0424_0800 - 0x0424_0fff
onchip_memorys1 0x0422		- 0×0423_ffff	0x0422_0000 - 0x0423_ffff
sramuas	0×0410_0000	- 0x041f_ffff	0x0410_0000 - 0x041f_fff



#### 5-2. Qsys インタコネクトのオプション設定

Interconnect Requirements タブで、システム全体や Qsys システム内のコンポーネントの要件などのオプション 設定を行うことができます。

System Contents 🕮 Address Map 🕮	😫 Interconnect Requirements 🛛 🛛					
Configure interconnect requirements for the system or an interface.						
System-wide Requirements						
Limit interconnect pipeline stages to: 📖 🗸						
Clock crossing adapter type:						
Clock crossing adapter type:   Hands	sha 👻					
Clock crossing adapter type: Hand:	sha 🔻					
All Requirements	sha 🔻					
All Requirements Identifier	sha ▼ Setting	Value				
All Requirements Identifier \$system	Setting Clock crossing adapter type	Value Handshake				
All Requirements Identifier \$system \$system	Setting Clock crossing adapter type Limit interconnect pipeline stages to	Value Handshake 1				
All Requirements Identifier Ssystem Ssystem Ssystem	Setting Clock crossing adapter type Limit interconnect pipeline stages to Enable ECC protection	Value Handshake 1 FALSE				

• Limit interconnect pipeline stages to

追加のレイテンシを犠牲にして fMAX(最大動作周波数)を増加させるために、各コマンドと応答のパスに挿入 することができるパイプライン・ステージの最大数を指定します。

<u>Clock crossing adapter type</u>

自動的に挿入されるクロック・クロッシング・アダプタのデフォルトに実装を指定します。

Handshake :単純なハンドシェーク・プロトコルを使用します。

FIFO :同期のため、デュアル・クロック FIFO を使用します。

Auto : バースト・リンク用 FIFO アダプタ、および他のすべてのリンクにはハンドシェイク・アダプタを 使用します。



#### 5-3. Qsys システムの生成

エラーがなくなったら、Qsys システムの生成を行います。エラーが残っているようであれば、エラー・メッセージ を確認してエラーを解消するようにしてください。

Qsys システムの生成を行うには、Qsys の Generate メニュー ⇒ Generate HDL を選択するか、右下の Generate HDL をクリックします。各オプションを設定したら、Generate をクリックします。

Generation		
Sunthacia		
Synthesis files are used to or	ampile the system in a Quartus Prime project	
Create HDL design files for s	withesis: Vaui	
Consta timina and maxim	an adirate for third each EDA such as a	
Create timing and resource	ce estimates for third-party EDA synthesis tools.	
✓ Create block symbol file	(bsf)	
Simulation		
The simulation model contain	s generated HDL files for the simulator, and may include simulation-only features.	
Simulation scripts for this co	moonent will be generated in a vendor-specific sub-directory in the specified output directory	
Follow the guidance in the ge and <i>ip-make-simscript</i> comm	nerated simulation scripts about how to structure your design's simulation scripts and how to use the <i>ip-setup-simul</i> and-line utilities to compile all of the files needed for simulating all of the IP in your design.	ation
Create simulation model:	None 👻	
Outrat Disastan		
Path:	Dr/work/o5.govo.1600/nice2.svotem	
	25 Work Co_days_room hose_aystein	

• <u>Create HDL design file for synthesis</u>

どの HDL で Qsys システムを生成するのかを指定します。Verilog / VHDL から指定できます。

• <u>Create block symbol file (.bsf)</u>

Quartus Prime の回路図エディタ用にシンボルを生成したい場合は、チェックを入れます。回路図でユーザ・ロジックと接続する場合に使用します。Output Directory で指定した先に、ファイルが生成されます。

Create simulation model

Qsys システムのシミュレーションを行いたい場合に、シミュレーション・モデルとシミュレーション用スクリプト・ファイルを生成してくれます。

Qsys システムの状態を保存していない場合は、まず保存されます。Close をクリックすると、Qsys システムの 生成へ進みます。

🐇 Save System Completed 💽				
All 这 🛆 🕕				
Info: E:/altera/16.0/ip/**/* matched 0 files in 0.00 seconds				
Info: Reading index E:¥altera¥16.0¥se¥quartus¥sopc_builder¥builtin.ipx				
Info: E:¥altera¥16.0¥se¥quartus¥sopc_builder¥builtin.ipx described 82 plugins, 0 p				
Info: E:/altera/16.0/se/quartus/sopc_builder/**/* matched 8 files in 0.01 seconds				
Info: Reading index E:¥altera¥16.0¥se¥quartus¥common¥librarian¥factories¥inde				
Info: E:¥altera¥16.0¥se¥quartus¥common¥librarian¥factories¥index.ipx describe				
Info: E:/altera/16.0/se/quartus/common/librarian/factories/**/* matched 4 fil				
🕕 Info: E:/altera/16.0/se/quartus/sopc_builder/bin/\$IP_IPX_PATH matched 1 files				
Info: E:¥altera¥16.0¥se¥quartus¥sopc_builder¥bin¥root_components.ipx describe				
Info: E:/altera/16.0/se/quartus/sopc_builder/bin/root_components.ipx matched				
• •				
Save System: completed successfully.				
Close				

エラーなく生成が終わったら、Close をクリックします。

🐇 Generate Completed
AII) 🔼 🛆 🕕
<ul> <li>Info: crosser: "mm_interconnect_0" instantiated altera_avalon_st_handshake_cloc</li> <li>Info: Reusing file D:/work/c5_qsys_1600/nios2_system/synthesis/submodules/¿</li> <li>Info: avalon_st_adapter: "mm_interconnect_0" instantiated altera_avalon_st_adapt</li> <li>Info: altera_asmi_parallel: generating top-level entity nios2_system_epoq_altera_asmi_paralle</li> <li>Info: altera_asmi_parallel: "altera_asmi_parallel" instantiated altera_asmi_paralle</li> <li>Info: altera_epcq_controller_core: "altera_epcq_controller_core" instantiated altera</li> <li>Info: nios2_system: Done "nios2_system" with 44 modules, 77 files</li> <li>Info: Finished: Create HDL design files for synthesis</li> </ul>
A Generate: completed with warnings.
Stop



#### 5-4. Qsys システムのテストベンチ・システム生成

Qsys の Generate メニュー  $\Rightarrow$  Generate Testbench System を選択すると、Qsys システムのシミュレーションを行う時のテストベンチ・システムとシミュレーション・モデルを生成してくれます。

Generation				
▼ Testbench System				
The testbench system is a new Qsys system that instantiates the original system, adding bus functional models to drive the top- Once generated, the bus functional models can interact with the system in the simulator.				
Create testbench Qsys system:	None			
Create testbench simulation mod	et Veril 🖵			
Output Directory				
Path:	D:/work/c5_gsys_1600/nios2_system			
Testbench:	D:/work/c5_qsys_1600/nios2_system/testbench/			
	Converte Converte			
	Generate Canc			

- Create testbench Qsys system
  - Standard, BFMs for standard Qsys Interconnect

Qsys システム全体をシミュレーションするためのスタンダードなテストベンチを生成します。しかし、 BFM は 32 ビットまでのアドレス幅をサポートしています。

• Simple, BFMs for clocks and resets

クロックとリセットのインタフェースのみのシンプルなテストベンチを生成します。

- ※ BFM:バス・ファンクション・モデル。バスの動作をシミュレーションで確認するためのモデル。
- <u>Create testbench simulation model</u>

どの HDL で テストベンチ・シミュレーション・モデルを生成するのかを指定します。Verilog / VHDL から指定 できます。

設定が終わったら、Generate をクリックします。そして、エラーなく生成が終わったら、Close をクリックします。

🔓 Generate Completed				
AII 😣 🛆 🕕				
Info: synopsys/vcsmx/vcsmx_setup.sh Info: Generating the following file(s) for NCSIM simulator in D:/work/c5_qsys_1600/n Info: cadence/cds_lib				
Info: cadence/hdl.var				
<ul> <li>Info: 49 .cds.lib files in cadence/cds_libs/ directory</li> <li>Info: 49 .cds.lib files in cadence/cds_libs/ directory</li> </ul>				
Info: aldec/rivierapro_setup.tcl				
<ul> <li>Into: For information on how to simulate your IP, see the explanatory comments in the sim Info: Finished: Create testbench Qsys system</li> </ul>				
Generate: completed with warnings.				
Stop				

#### 5-5. Qsys システムのインスタンスを含んだテンプレート

Qsys の Generate メニュー ⇒ Show Instantiation Template を選択すると、Qsys システムの 1つ上の階層に 記述するインスタンス用のテンプレートが表示されます。これをコピーして、ユーザ・ロジックの HDL に貼り付け て編集すれば、簡単にユーザ・ロジックと Qsys システムを接続することができます。Verilog / VHDL を選択する ことができます。

🚡 Instantiation Template						
You can copy the example HDL below to declare an instance of <b>nios2 system</b> .						
HDL Language: Veril HDL Language: Verilog Example HDL VHDL nios2_system u0 ( .clk_clk ( <connected-to-clk_clk>), .ext_sram_bus_out_tcm_chipselect_n_out (<connected-to-ext_sram_bus_out_tcm_chipsele (<connected-to-ext_sram_bus_out_tcm_byteenable_n_out="" (<connected-to-ext_sram_bus_out_tcm_outputenable_n_out="" )<="" .ext_sram_bus_out_tcm_byteenable_n_out="" td=""></connected-to-ext_sram_bus_out_tcm_chipsele></connected-to-clk_clk>						
<pre>.ext_sram_bus_out_tcm_write_n_out (<connected-to-ext_sram_bus_out_tcm_write_n_i (<connected-to-ext_sram_bus_out_sram_begintranside_n_out)="" (<connected-to-ext_sram_bus_out_tcm_begintranside_n_out)="" (<connected-to-ext_sram_bus_out_tcm_data_out)="" (<connected-to-ext_sram_bus_out_tcm_reset_n_i="" .ext_sram_bus_out_sram_bus_o<="" .ext_sram_bus_out_tcm_begintransfer_n_out="" .ext_sram_bus_out_tcm_begintranside_n_out)="" .ext_sram_bus_out_tcm_data_out="" .ext_sram_bus_out_tcm_reset_n_out="" td=""></connected-to-ext_sram_bus_out_tcm_write_n_i></pre>						
);						
Copy Close						

#### 6. Quartus Prime での各種設定

Qsys システムの生成が完了したら、ユーザ・ロジックと接続して FPGA デザインを完成させます。FPGA デザ インが完成したらコンパイルを行いますが、その前に必要な設定を行います。この資料では、Qsys システムに関 係する部分のみ、説明します。

#### 6-1. Qsys システム用ファイルの登録

Quartus Prime の Assignments メニュー ⇒ Settings の Files カテゴリで、生成した Qsys システムの .qip フ ァイルを登録します。.qip ファイルは生成した Qsys システムの関連情報が格納されており、.qsys ファイルを生成 したディレクトリの下位ディレクトリに生成されています。このファイルの登録を行わないと、コンパイルでエラーと なります。

例)<.qsys ファイル名と同名のディレクトリ>¥synthesis

egory:				Device/Boar
Conordi	Files			
Files Libraries IP Settings	Select the design files you want to include in the project. Click Add All to add all design files in the project directory to the project.			
IP Catalog Search Locations Design Templates	File name: nios2_system/sy	nthesis/nios2_system.qip		Add
Operating Settings and Conditions Voltage	٩			× Add All
Temperature	File Name	Туре	Library Desig	gn Enty Remove
Compilation Process Settings Incremental Compilation     EDA Tool Settings     Design Entry/Synthesis     Simulation	top.bdf	Block Diagram/Schem	atic File <non< td=""><td>le&gt;</td></non<>	le>
				Down
Board-Level				Properties
gory:				Device/Bo
General	Files			
Files Libraries IP Settings	Select the design files you w project.	ant to include in the project. Click Add All to	add all n files in th	e project directory to th
IP Catalog Search Locations Design Templates	File name:			Add
Operating Settings and Conditions Voltage	•			X Add All
Temperature	File Name	Туре	Library Desi	Remove
Incremental Compilation	nios2_system/synthesis	s/nios2_system.qip IP Variation File (.qip)	<nor< td=""><td>le&gt; Up</td></nor<>	le> Up
EDA Tool Settings				Down
Simulation				

#### 6-2. .sdc ファイルの登録

Qsys システム内のタイミング制約用の .sdc ファイルは上記の .qip ファイルを登録すれば、Qsys システム部 分の .sdc ファイルは自動で登録されます。.sdc ファイルが登録されていない場合は、Quartus Prime の Assignments メニュー ⇒ Settings の TimeQuest Timing Analyzer カテゴリで必要な .sdc ファイルを手動で登録 してください。

ユーザ・ロジック部分の .sdc ファイルはユーザが作成して、手動で登録してください。



### <u> 改版履歴</u>

Revision	年月	概要
1	2016年10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。
   株式会社アルティマ ホームページ: http://www.altima.co.jp
   技術情報サイト アルティマ技術データベース: 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。