

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

Ver.16

# Quartus Prime ガイド

## Qsys システム統合ツールの使い方

### 目次

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 システムの保存 .....	7
2-4. Qsys の終了 .....	7
3. コンポーネントの追加 .....	8
3-1. コンポーネントの検索 .....	8
3-2. コンポーネントの追加と各種設定 .....	8
3-3. コンポーネントの並べ替え .....	10
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 .....	12
4-1-3. 出力ポート .....	12
4-1-4. 各コンポーネント間のクロック接続 .....	12
4-2. マスタとスレーブ間の接続 .....	17
4-3. 割り込み .....	18
4-3-1. 割り込みの接続 .....	18
4-3-2. 割り込みの設定 .....	18
4-4. Qsys システムの外部に出すピン .....	19
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 .....	20
5-2. Qsys インタコネクットのオプション設定 .....	21
5-3. Qsys システムの生成 .....	22
5-4. Qsys システムのテストベンチ・システム生成 .....	24
5-5. Qsys システムのインスタンスを含んだテンプレート .....	25

# Quartus Prime ガイド

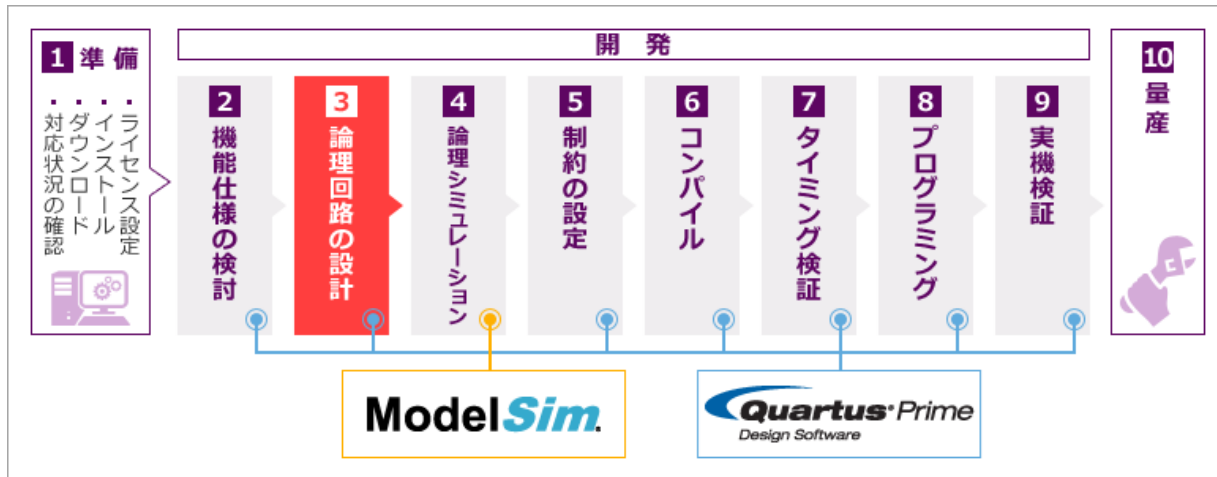
## Qsys システム統合ツールの使い方

---

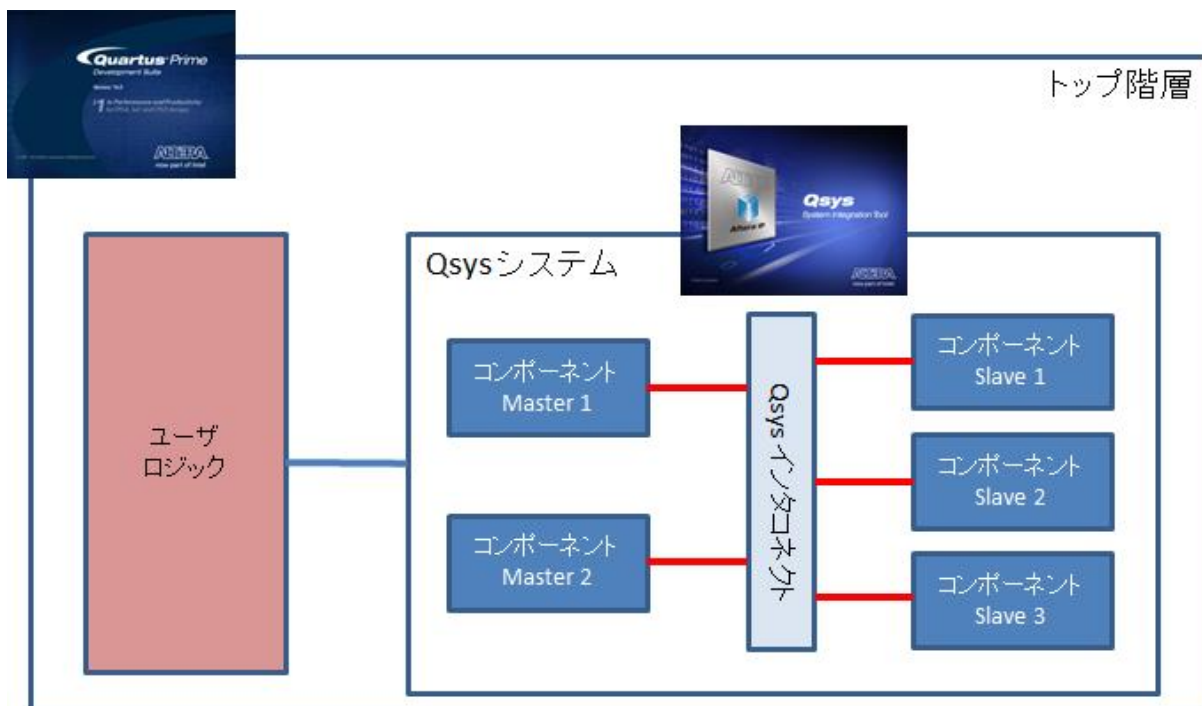
6. Quartus Prime での各種設定.....	26
6-1. Qsys システム用ファイルの登録 .....	26
6-2. .sdc ファイルの登録 .....	26
改版履歴 .....	27

## 1. はじめに

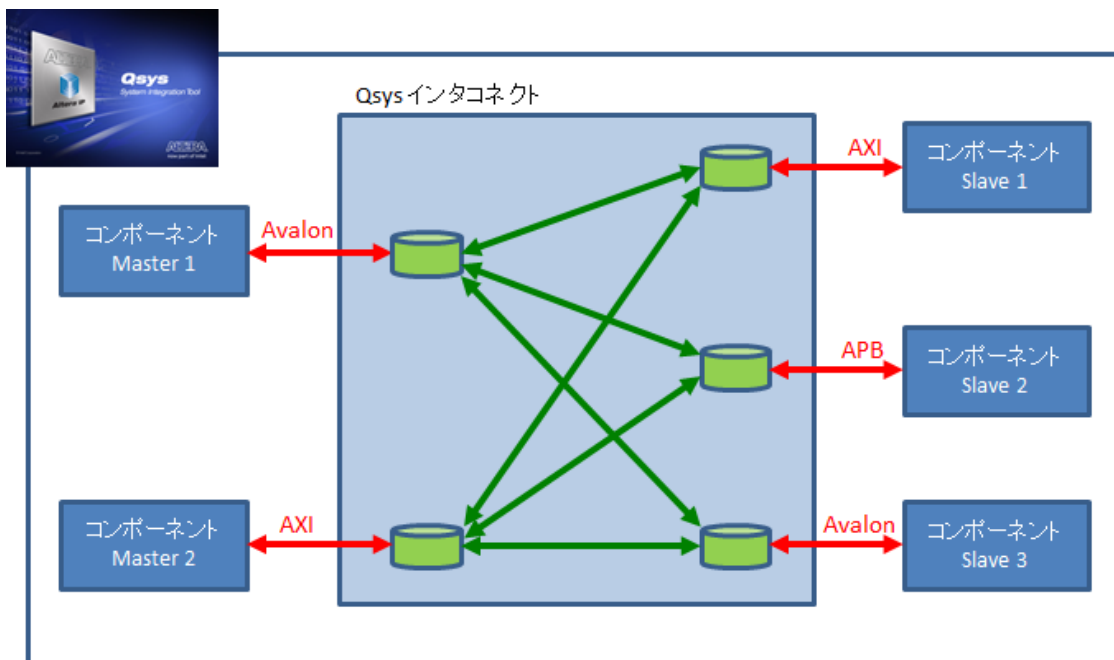
この資料は、Quartus<sup>®</sup> Prime 開発ソフトウェア(以下、Quartus Prime)を使用するユーザ向けの資料です。FPGA / CPLD の開発フローの中で、主に下図の赤枠内の開発フェーズで非常に参考になる資料です。



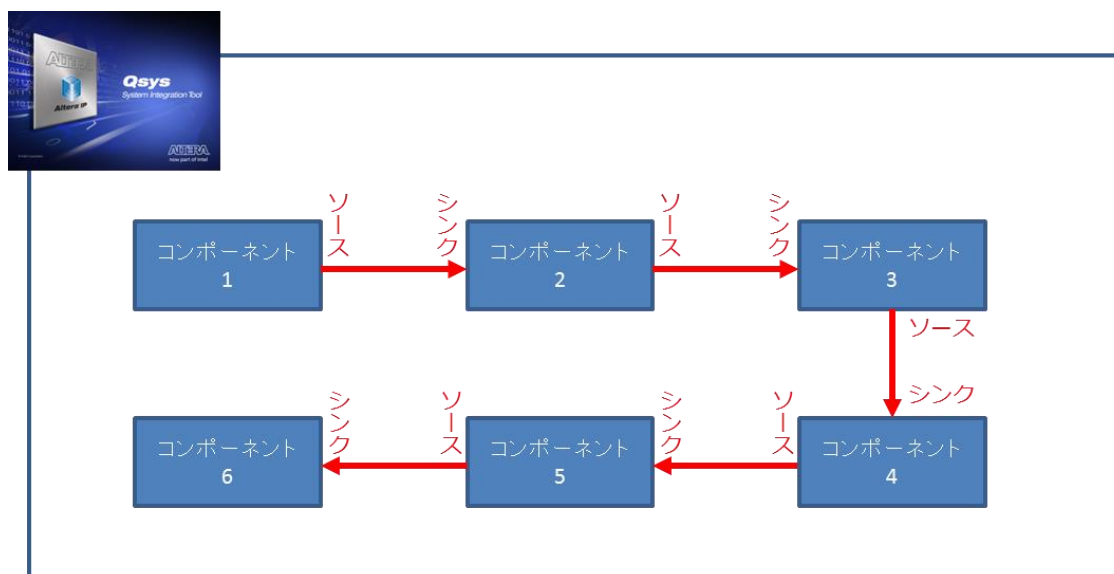
Qsys システム統合ツール(以下、Qsys)は、各コンポーネント(Intellectual Property (IP) ファンクションやサブシステム、ユーザ・ロジックなど)を接続するインタコネク・ロジック(アドレス/データ・バスの接続やバス幅整合ロジック、アドレス・デコーダ・ロジック、アービトレーション・ロジックなど)を自動的に生成して、FPGA デザイン・プロセスの時間と労力を大幅に削減します。Qsys は FPGA に最適化されたネットワーク・オン・チップ(NoC)の新しい技術で開発されており 前世代ツールの SOPC Builder と比べて、性能の向上やデザインの再利用のしやすさ、そ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

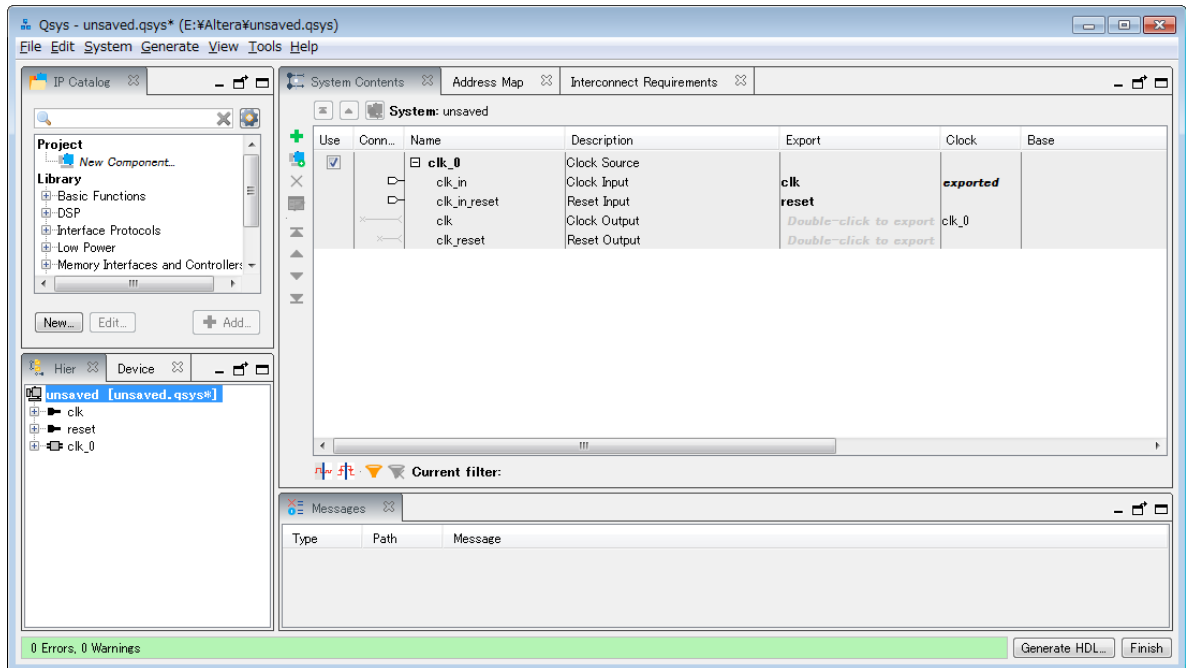


Nios<sup>®</sup> II プロセッサ(以下、Nios II)や FPGA デバイスに内蔵された ARM<sup>®</sup> コア内蔵の SoC FPGA を使用する場などに限らず、最近では Qsys を使用するケースが多くあるため、この資料では Qsys の簡単な使い方を説明します。

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. Qsys の基本操作

Qsys は Quartus Prime から起動します。Quartus Prime の Tools メニュー ⇒ Qsys を選択するか  ツールバーをクリックしてください。



### 2-1. Qsys システムの新規作成

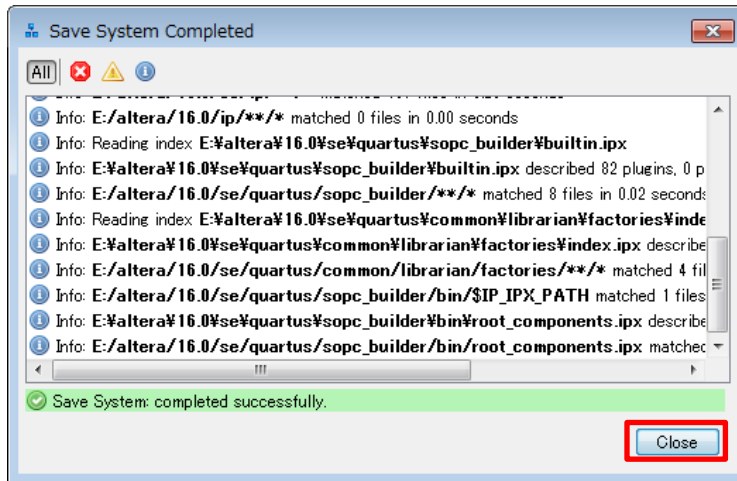
上図が Qsys を新規に起動した状態です。

### 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 ⇒ Open を選択して、該当の .qsys ファイルを選択してください。

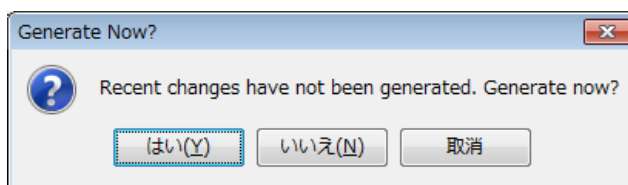
### 2-3. Qsys システムの保存

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。)



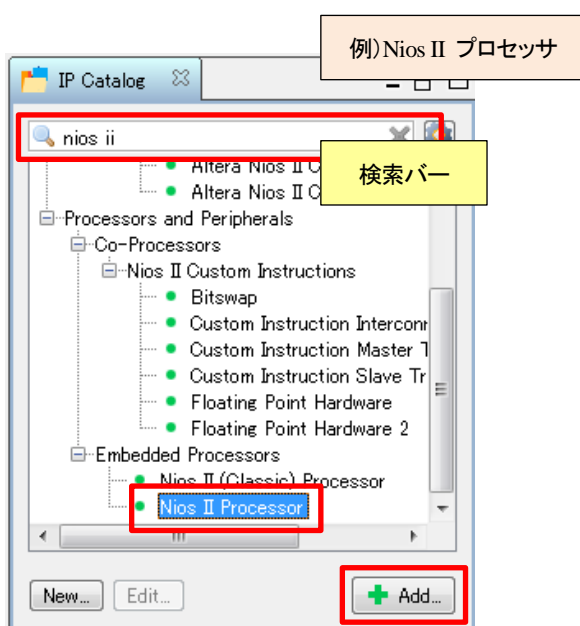
### 3. コンポーネントの追加

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。



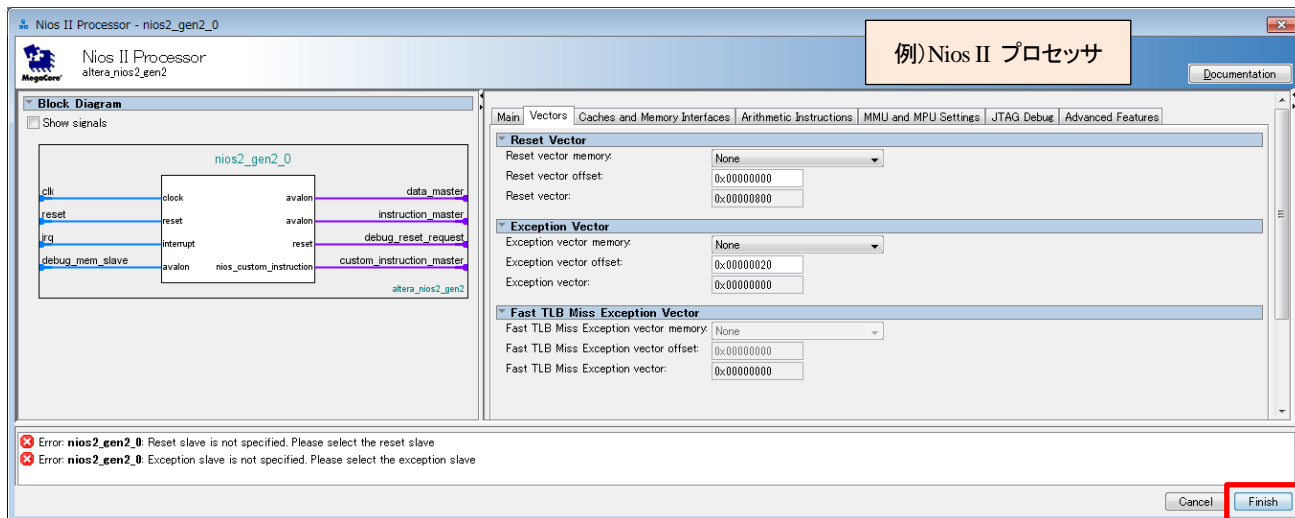
#### 3-2. コンポーネントの追加と各種設定

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。)

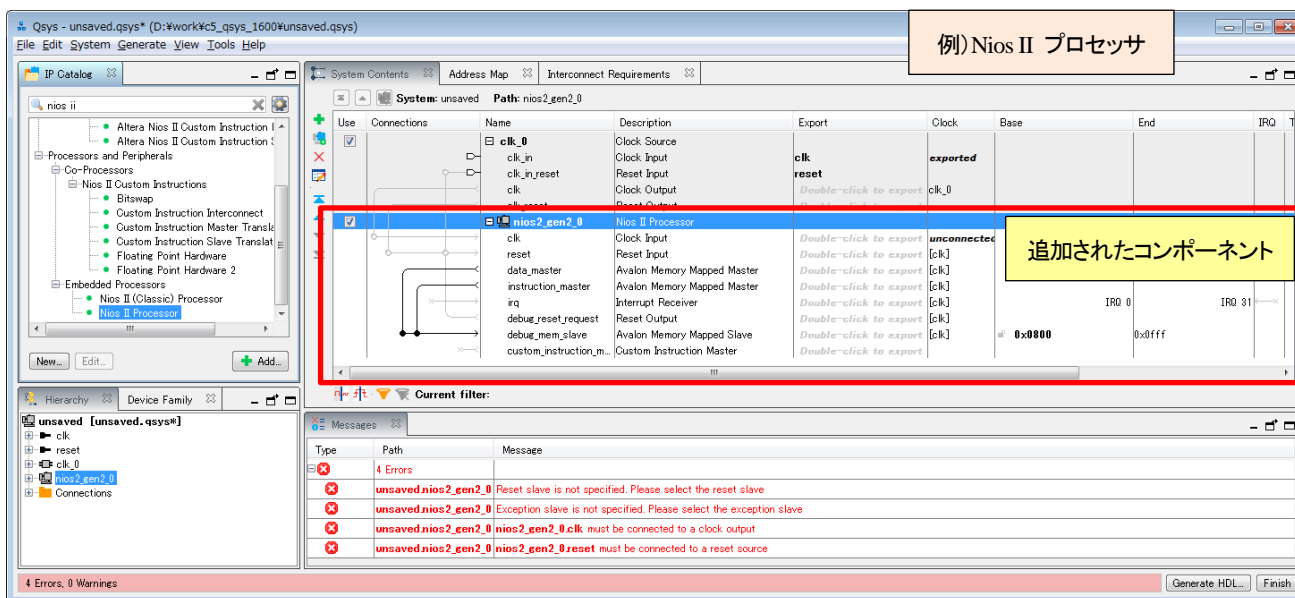
※ ここでは、各コンポーネントの設定画面の詳細については説明しません。設定の詳細については、各コンポーネントのドキュメントを参照してください。



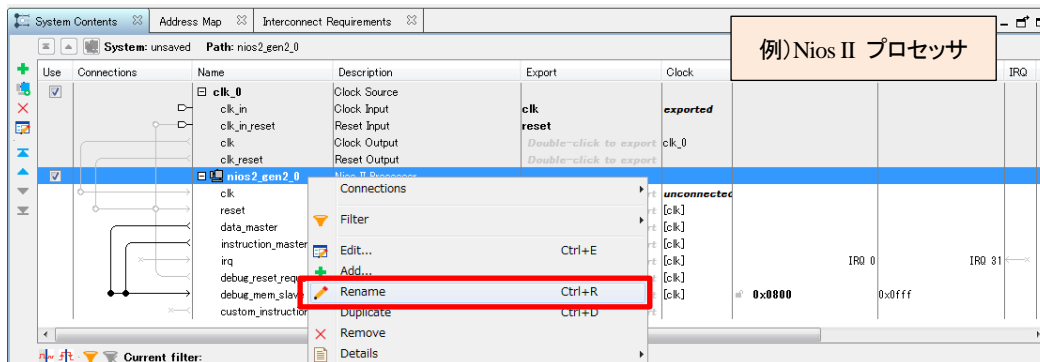
必要な設定ができたなら、Finish をクリックします。



すると、Qsys システム内にコンポーネントが追加されます。







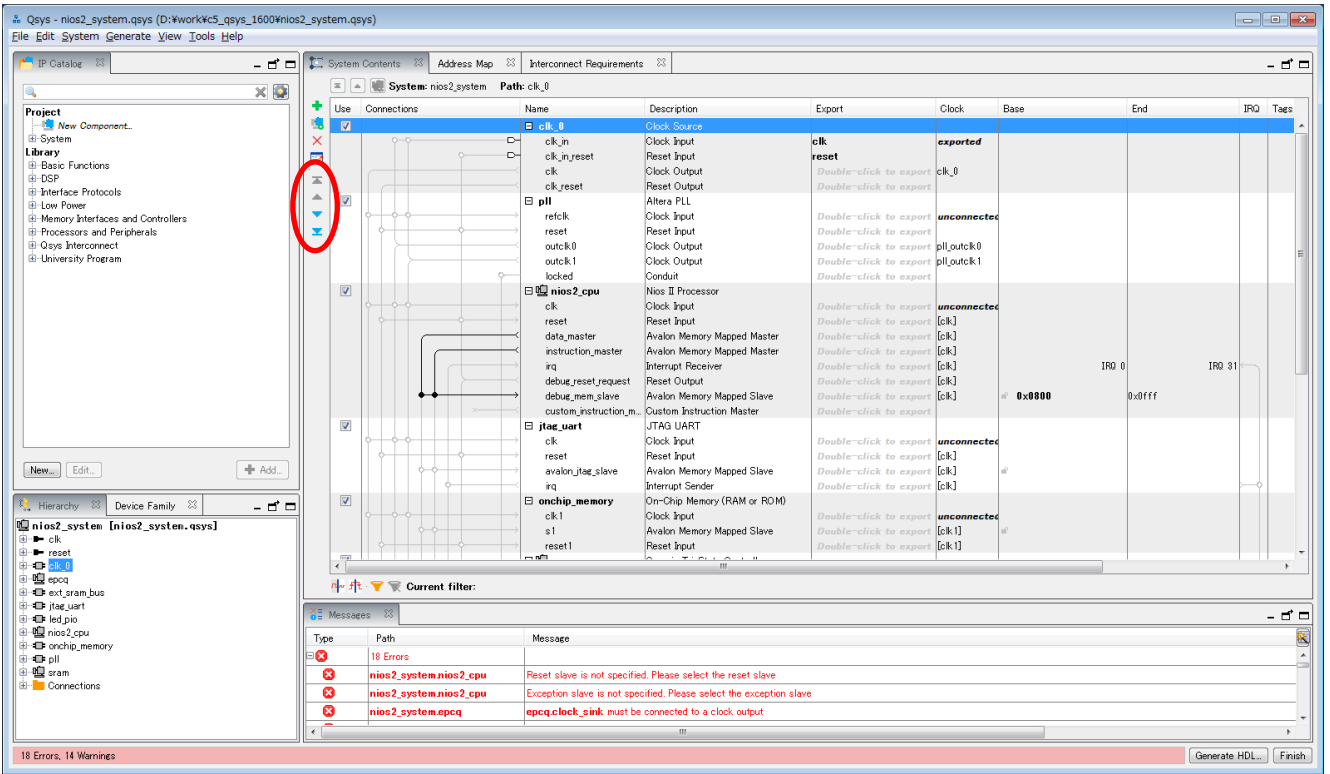
追加したコンポーネントの名前をデフォルトから変更したい場合は、該当のコンポーネントをハイライトして、右クリック ⇒ Rename を選択して、名前を修正してください。



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。

### 3-3. コンポーネントの並べ替え

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。



Use	Connections	Name	Description	Export	Clock	Base	End	IRQ	Tags
<input checked="" type="checkbox"/>		clk_0	Clock Source	clk	exported				
<input checked="" type="checkbox"/>		clk_in	Reset Input	reset					
<input checked="" type="checkbox"/>		clk_in_reset	Reset Input						
<input checked="" type="checkbox"/>		clk	Clock Output						
<input checked="" type="checkbox"/>		clk_reset	Reset Output						
<input checked="" type="checkbox"/>		pll	Altera PLL						
<input checked="" type="checkbox"/>		retclk	Clock Input						
<input checked="" type="checkbox"/>		reset	Reset Input						
<input checked="" type="checkbox"/>		outclk0	Clock Output						
<input checked="" type="checkbox"/>		outclk1	Clock Output						
<input checked="" type="checkbox"/>		locked	Conduit						
<input checked="" type="checkbox"/>		nios2_cpu	Nios II Processor						
<input checked="" type="checkbox"/>		clk	Clock Input						
<input checked="" type="checkbox"/>		reset	Reset Input						
<input checked="" type="checkbox"/>		data_master	Avakon Memory Mapped Master						
<input checked="" type="checkbox"/>		instruction_master	Avakon Memory Mapped Master						
<input checked="" type="checkbox"/>		irq	Interrupt Receiver					IRQ 0	IRQ 31
<input checked="" type="checkbox"/>		debug_reset_request	Reset Output						
<input checked="" type="checkbox"/>		debug_mem_slave	Avakon Memory Mapped Slave			0x0800	0x0fff		
<input checked="" type="checkbox"/>		custom_instruction_m	Custom Instruction Master						
<input checked="" type="checkbox"/>		jtag_uart	JTAG UART						
<input checked="" type="checkbox"/>		clk	Clock Input						
<input checked="" type="checkbox"/>		reset	Reset Input						
<input checked="" type="checkbox"/>		avalon_jtag_slave	Avakon Memory Mapped Slave						
<input checked="" type="checkbox"/>		irq	Interrupt Sender						
<input checked="" type="checkbox"/>		onchip_memory	On-Chip Memory (RAM or ROM)						
<input checked="" type="checkbox"/>		clk1	Clock Input						
<input checked="" type="checkbox"/>		s1	Avakon Memory Mapped Slave						
<input checked="" type="checkbox"/>		reset1	Reset Input						

Type	Path	Message
18 Errors		
	nios2_system.nios2_cpu	Reset slave is not specified. Please select the reset slave
	nios2_system.nios2_cpu	Exception slave is not specified. Please select the exception slave
	nios2_system.epcq	epcq.clock_sink must be connected to a clock output

## 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ー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。

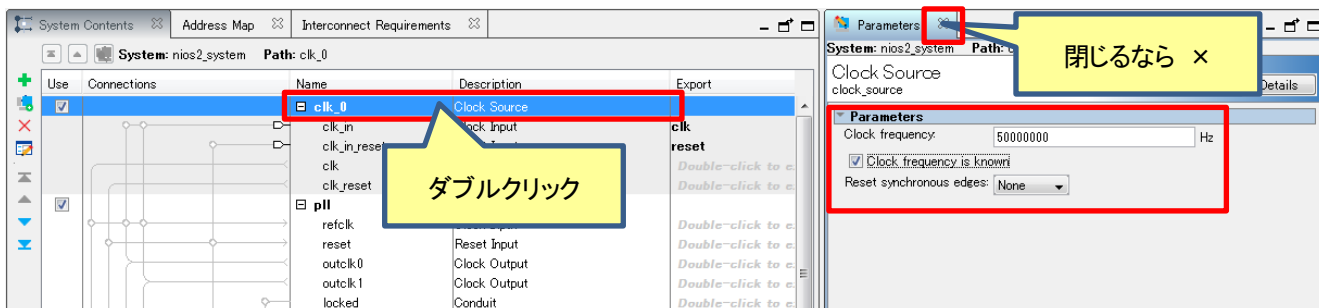
※ Nios II プロセッサの Reset Vector や Exception Vector などは、Qsys システム上の接続を行わないと設定ができなかったり、エラーが消えなかったりします。これは一旦、エラーが残っている状態で Qsys システムへの Nios II プロセッサの追加を済ませた後に Qsys システム内の接続を適切に行うことで、設定ができるようになり、エラーが消えます。この資料では詳細な説明を省略していますので、詳しくは各コンポーネントのドキュメントを参照してください。

### 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。



#### ◆ 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)がデフォルトで 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
<input checked="" type="checkbox"/>		clk_0	Clock Source		
		clk_in	Clock Input	<b>clk</b>	<b>exported</b>
		clk_in_reset	Reset Input	<b>reset</b>	
		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
<input checked="" type="checkbox"/>		clk_0	Clock Source		
		clk_in	Clock Input	<b>clk</b>	<b>exported</b>
		clk_in_reset	Reset Input	<b>reset</b>	
		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ため、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
<input checked="" type="checkbox"/>		clk_0	Clock Source		
		clk_in	Clock Input	<b>clk</b>	<b>exported</b>
		clk_in_reset	Reset Input	<b>reset</b>	
		clk	Clock Output	Double-click to export	clk_0
		clk_reset	Reset Output	Double-click to export	
<input checked="" type="checkbox"/>		pll	Altera PLL		
		refclk	Clock Input	Double-click to export	<b>unconnected</b>
		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	

Use	Connections	Name	Description	Export	Clock	
<input checked="" type="checkbox"/>		<b>clk_0</b>	Clock Source			
		clk_in	Clock Input	<b>clk</b>	<b>exported</b>	
		clk_in_reset	Reset Input	<b>reset</b>		
		clk	Clock Output	<i>Double-click to export</i>	clk_0	
		clk_reset	Reset Output	<i>Double-click to export</i>		
<input checked="" type="checkbox"/>			<b>pll</b>	Altera PLL		
		refclk	Clock Input	<i>Double-click to export</i>	clk_0	
		reset	Reset Input	<i>Double-click to export</i>		
		outclk0	Clock Output	<i>Double-click to export</i>	pll_outclk0	
		outclk1	Clock Output	<i>Double-click to export</i>	pll_outclk1	
	locked	Conduit	<i>Double-click to export</i>			

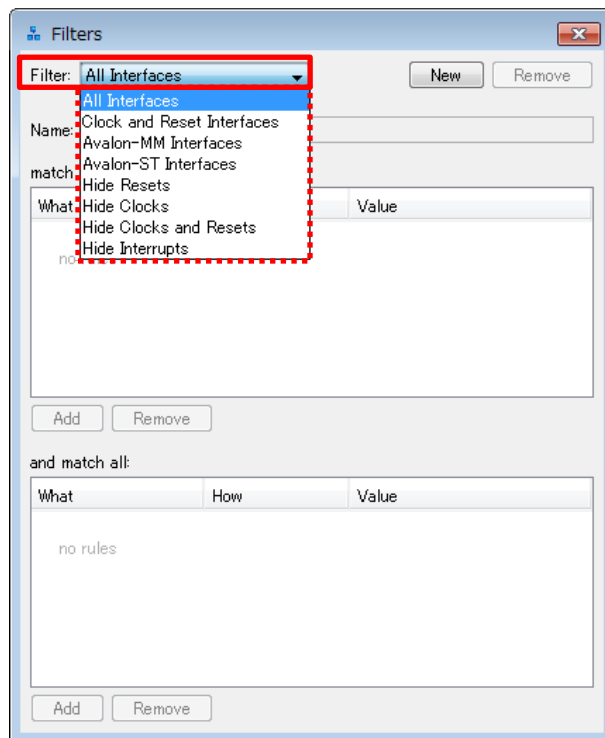
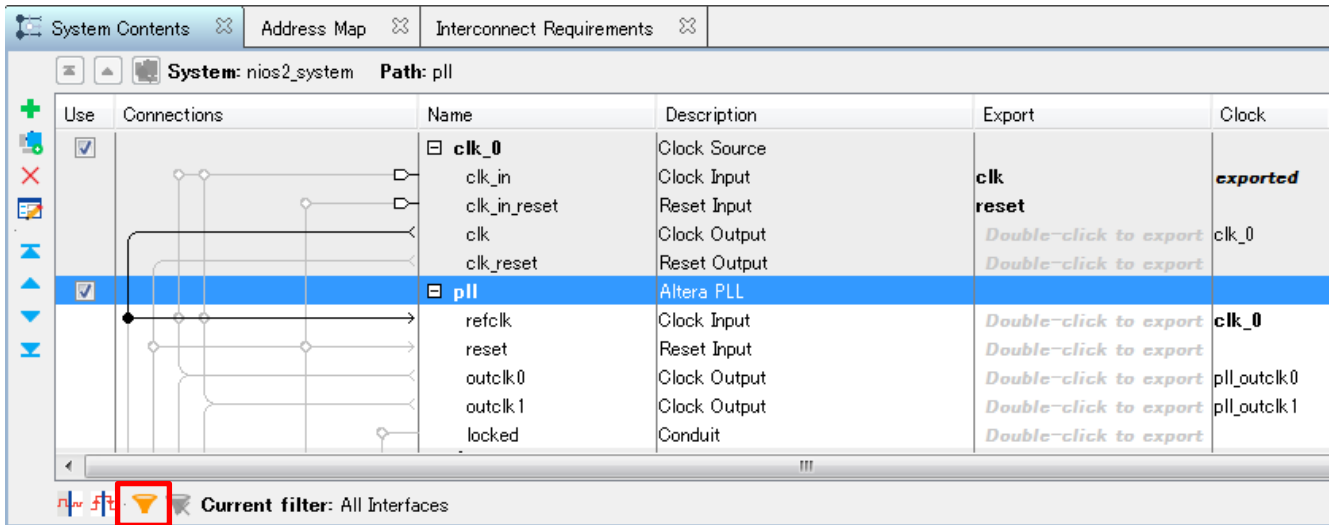
次に、PLL で生成した 2 種類のクロックをシステムの仕様に応じて各コンポーネントのクロック入力へ接続します。ここで、表示のフィルタ機能を使用してクロックとリセット関係のみの表示 (Clock and Reset Interfaces) にすると、接続がやりやすくなります。

Use	Connections	Name	Description	Export	Clock	
<input checked="" type="checkbox"/>		<b>clk_0</b>	Clock Source			
		clk_in	Clock Input	<b>clk</b>	<b>exported</b>	
		clk_in_reset	Reset Input	<b>reset</b>		
		clk	Clock Output	<i>Double-click to export</i>	clk_0	
		clk_reset	Reset Output	<i>Double-click to export</i>		
<input checked="" type="checkbox"/>			<b>pll</b>	Altera PLL		
		refclk	Clock Input	<i>Double-click to export</i>	clk_0	
		reset	Reset Input	<i>Double-click to export</i>		
		outclk0	Clock Output	<i>Double-click to export</i>	pll_outclk0	
		outclk1	Clock Output	<i>Double-click to export</i>	pll_outclk1	
<input checked="" type="checkbox"/>		<b>nios2_cpu</b>	Nios II Processor			
	clk	Clock Input	<i>Double-click to export</i>	pll_outclk0		
	reset	Reset Input	<i>Double-click to export</i>	[clk]		
	debug_reset_request	Reset Output	<i>Double-click to export</i>	[clk]		
<input checked="" type="checkbox"/>		<b>jtag_uart</b>	JTAG UART			
	clk	Clock Input	<i>Double-click to export</i>	pll_outclk0		
	reset	Reset Input	<i>Double-click to export</i>	[clk]		
<input checked="" type="checkbox"/>		<b>onchip_memory</b>	On-Chip Memory (RAM or ROM)			
	clk1	Clock Input	<i>Double-click to export</i>	pll_outclk0		
	reset1	Reset Input	<i>Double-click to export</i>	[clk 1]		
<input checked="" type="checkbox"/>		<b>sram</b>	Generic Tri-State Controller			
	clk	Clock Input	<i>Double-click to export</i>	pll_outclk0		
	reset	Reset Input	<i>Double-click to export</i>	[clk]		
<input checked="" type="checkbox"/>		<b>ext_sram_bus</b>	Tri-State Conduit Bridge			
	clk	Clock Input	<i>Double-click to export</i>	pll_outclk0		
	reset	Reset Input	<i>Double-click to export</i>	[clk]		
<input checked="" type="checkbox"/>		<b>epcq</b>	Altera Serial Flash Controller			
	clock_sink	Clock Input	<i>Double-click to export</i>	pll_outclk1		
	reset	Reset Input	<i>Double-click to export</i>	[clock_sink]		
<input checked="" type="checkbox"/>		<b>led_pio</b>	PIO (Parallel I/O)			
	clk	Clock Input	<i>Double-click to export</i>	pll_outclk0		
	reset	Reset Input	<i>Double-click to export</i>	[clk]		

表示のフィルタ: Clock and Reset Interface

【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。



### 【補足②:接続の方法】

先ほど、Connections 欄で ○ 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。

#### ◆ クロック入力の Clock 欄で選択する方法

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
<input checked="" type="checkbox"/>		clk_0	Clock Source		
		clk_in	Clock Input	clk	exported
		clk_in_reset	Reset Input	reset	
		clk	Clock Output	clk	clk_0
		clk_reset	Reset Output	reset	
<input checked="" type="checkbox"/>		pll	Altera PLL		
		refclk	Clock Input	refclk	clk_0
		reset	Reset Input	reset	
		outclk0	Clock Output	pll_outclk0	pll_outclk0
		outclk1	Clock Output	pll_outclk1	pll_outclk1
		locked	Conduit		
<input checked="" type="checkbox"/>		nios2_cpu	Nios II Processor		
		clk	Clock Input	pll_outclk0	pll_outclk1
		reset	Reset Input	reset	
		data_master	Avalon Memory Mapped Master		
		instruction_master	Avalon Memory Mapped Master		
		irq	Interrupt Receiver		[clk]
		debug_reset_request	Reset Output		[clk]
		debug_mem_slave	Avalon Memory Mapped Slave		[clk]
		custom_instruction_m...	Custom Instruction Master		

#### ◆ 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。

Use	Connections	Name	Description	Export	Clock	Base	End
<input checked="" type="checkbox"/>		clk_0	Clock Source				
		clk_in	Clock Input	clk	exported		
		clk_in_reset	Reset Input	reset			
		clk	Clock Output	clk	clk_0		
		clk_reset	Reset Output	reset			
<input checked="" type="checkbox"/>		pll	Altera PLL				
		refclk	Clock Input	refclk	clk_0		
		reset	Reset Input	reset			
		outclk0	Clock Output	pll_outclk0			
		outclk1	Clock Output	pll_outclk1			
		locked	Conduit				
<input checked="" type="checkbox"/>		nios2_cpu	Nios II Processor				
		clk	Clock Input	pll_outclk0	pll_outclk1		
		reset	Reset Input	reset			
		data_master	Avalon Memory Mapped Master				
		instruction_master	Avalon Memory Mapped Master				
		irq	Interrupt Receiver				[clk]
		debug_reset_request	Reset Output				[clk]
		debug_mem_slave	Avalon Memory Mapped Slave				[clk]
		custom_instruction_m...	Custom Instruction Master				
<input checked="" type="checkbox"/>			jtag_uart	JTAG UART			

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
<input checked="" type="checkbox"/>		<b>clk_0</b>	Clock Source		
		clk_in	Clock Input	<b>clk</b>	<b>exported</b>
		clk_in_reset	Reset Input	<b>reset</b>	
		clk	Clock Output	<i>Double-click to export</i>	clk_0
		clk_reset	Reset Output		
<input checked="" type="checkbox"/>		<b>pll</b>	Altera PLL		
		refclk	Clock Input	<i>Double-click to export</i>	clk_0
		reset	Reset Input	<i>Double-click to export</i>	
		outclk0	Clock Output	<i>Double-click to export</i>	pll_outclk0
		outclk1	Clock Output	<i>Double-click to export</i>	pll_outclk1
<input checked="" type="checkbox"/>		<b>nios2_cpu</b>	Nios II Processor		
		clk	Clock Input	<i>Double-click to export</i>	pll_outclk0
		reset	Reset Input	<i>Double-click to export</i>	[clk]
		debug_reset_request	Reset Output	<i>Double-click to export</i>	[clk]
<input checked="" type="checkbox"/>		<b>jtag_uart</b>	JTAG UART		
		clk	Clock Input	<i>Double-click to export</i>	pll_outclk0
		reset	Reset Input	<i>Double-click to export</i>	[clk]
<input checked="" type="checkbox"/>		<b>onchip_memory</b>	On-Chip Memory (RAM or ROM)		
		clk1	Clock Input	<i>Double-click to export</i>	pll_outclk0
		reset1	Reset Input	<i>Double-click to export</i>	[clk1]
<input checked="" type="checkbox"/>		<b>sram</b>	Generic Tri-State Controller		
		clk	Clock Input	<i>Double-click to export</i>	pll_outclk0
		reset	Reset Input	<i>Double-click to export</i>	[clk]
<input checked="" type="checkbox"/>		<b>ext_sram_bus</b>	Tri-State Conduit Bridge		
		clk	Clock Input	<i>Double-click to export</i>	pll_outclk0
		reset	Reset Input	<i>Double-click to export</i>	[clk]
<input checked="" type="checkbox"/>		<b>epcq</b>	Altera Serial Flash Controller		
		clock_sink	Clock Input	<i>Double-click to export</i>	pll_outclk1
		reset	Reset Input	<i>Double-click to export</i>	[clock_sink]
<input checked="" type="checkbox"/>		<b>led_pio</b>	PIO (Parallel I/O)		
		clk	Clock Input	<i>Double-click to export</i>	pll_outclk0
		reset	Reset Input	<i>Double-click to export</i>	[clk]

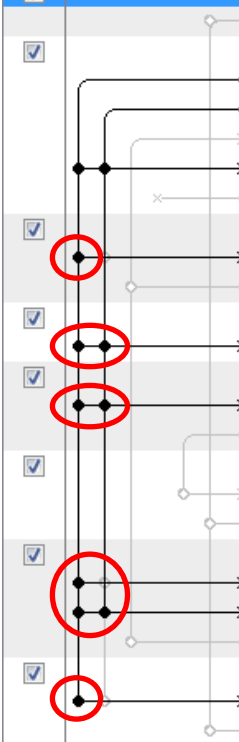
表示のフィルタ: Clock and Reset Interface



## 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
<input checked="" type="checkbox"/>		<b>pll</b>	Altera PLL		clk_0
<input checked="" type="checkbox"/>		locked	Conduit		
<input checked="" type="checkbox"/>		<b>nios2_cpu</b>	Nios II Processor	表示のフィルタ: Hide Clock and Reset	
		data_master	Avalon Memory Mapped Master	Double-click to export	pll_outclk0 [clk]
		instruction_master	Avalon Memory Mapped Master	Double-click to export	[clk]
		irq	Interrupt Receiver	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	[clk]
<input checked="" type="checkbox"/>		<b>jtag_uart</b>	JTAG UART		[clk]
		avalon_jtag_slave	Avalon Memory Mapped Slave	Double-click to export	pll_outclk0 [clk]
		irq	Interrupt Sender	Double-click to export	[clk]
<input checked="" type="checkbox"/>		<b>onchip_memory</b>	On-Chip Memory (RAM or ROM)		[clk 1]
		s1	Avalon Memory Mapped Slave	Double-click to export	pll_outclk0 [clk]
<input checked="" type="checkbox"/>		<b>sram</b>	Generic Tri-State Controller		[clk]
		uas	Avalon Memory Mapped Slave	Double-click to export	pll_outclk0 [clk]
		tcm	Tristate Conduit Master	Double-click to export	[clk]
<input checked="" type="checkbox"/>		<b>ext_sram_bus</b>	Tri-State Conduit Bridge		[clk]
		tcs	Tristate Conduit Slave	Double-click to export	pll_outclk0 [clk]
		out	Conduit	Double-click to export	
<input checked="" type="checkbox"/>		<b>epcq</b>	Altera Serial Flash Controller		[clock_sink]
		avl_csr	Avalon Memory Mapped Slave	Double-click to export	pll_outclk1 [clock_sink]
		avl_mem	Avalon Memory Mapped Slave	Double-click to export	[clock_sink]
		interrupt_sender	Interrupt Sender	Double-click to export	[clock_sink]
<input checked="" type="checkbox"/>		<b>led_pio</b>	PIO (Parallel I/O)		[clk]
		s1	Avalon Memory Mapped Slave	Double-click to export	pll_outclk0 [clk]
		external_connection	Conduit	Double-click to export	

### 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
<input checked="" type="checkbox"/>		<b>pll</b> locked	Altera PLL Conduit	<i>Double-click to export</i>	clk_0				
<input checked="" type="checkbox"/>		<b>nios2_cpu</b> data_master instruction_master	Nios II Processor Avalon Memory Mapped Master Avalon Memory Mapped Master	<i>Double-click to export</i> <i>Double-click to export</i>	[clk] [clk]				
<input checked="" type="checkbox"/>		<b>irq</b>	Interrupt Receiver	<i>Double-click to export</i>	[clk]		IRQ 0	IRQ 31	
<input checked="" type="checkbox"/>		debug_mem_slave custom_instruction_m...	Avalon Memory Mapped Slave Custom Instruction Master	<i>Double-click to export</i> <i>Double-click to export</i>	[clk]	# 0x0000_0800	0x0000_0fff		
<input checked="" type="checkbox"/>		<b>jtag_uart</b> avalon_jtag_slave irq	JTAG UART Avalon Memory Mapped Slave Interrupt Sender	<i>Double-click to export</i> <i>Double-click to export</i>	[clk] [clk]	# 0x0000_0000	0x0000_0007		<input type="text" value="0"/>
<input checked="" type="checkbox"/>		<b>onchip_memory</b> s1	On-Chip Memory (RAM or ROM) Avalon Memory Mapped Slave	<i>Double-click to export</i>	[clk1]	# 0x0000_0000	0x0001_ffff		<input type="text" value="0"/>
<input checked="" type="checkbox"/>		<b>sram</b> uas tcm	Generic Tri-State Controller Avalon Memory Mapped Slave Tristate Conduit Master	<i>Double-click to export</i> <i>Double-click to export</i>	[clk] [clk]	# 0x0000_0000	0x000f_ffff		
<input checked="" type="checkbox"/>		<b>ext_sram_bus</b> tcs out	Tri-State Conduit Bridge Tristate Conduit Slave Conduit	<i>Double-click to export</i> <i>Double-click to export</i>	[clk]				
<input checked="" type="checkbox"/>		<b>epcq</b> avl_csr avl_mem interrupt_sender	Altera Serial Flash Controller Avalon Memory Mapped Slave Avalon Memory Mapped Slave Interrupt Sender	<i>Double-click to export</i> <i>Double-click to export</i> <i>Double-click to export</i>	[clock_sink] [clock_sink] [clock_sink]	# 0x0000_0000 # 0x0000_0000	0x0000_001f 0x01ff_ffff		<input type="text" value="0"/>
<input checked="" type="checkbox"/>		<b>led_pio</b> s1 external_connection	PIO (Parallel I/O) Avalon Memory Mapped Slave Conduit	<i>Double-click to export</i> <i>Double-click to export</i>	[clk] [clk]	# 0x0000_0000	0x0000_000f		

#### 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。

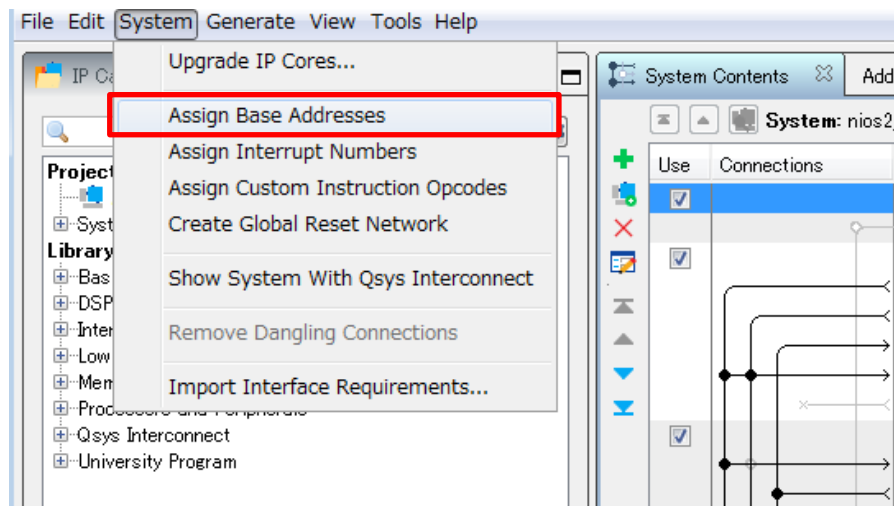
Use	Connections	Name	Description	Export	Clock	E
<input checked="" type="checkbox"/>		pll	Altera PLL		clk_0	
		locked	Conduit	<b>pll_locked</b>		
<input checked="" type="checkbox"/>		nios2_cpu	Nios II Processor		[clk]	
		data_master	Avalon Memory Mapped Master	Double-click to export	pll_outclk0	
		instruction_master	Avalon Memory Mapped Master	Double-click to export	[clk]	
		irq	Interrupt Receiver	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		
<input checked="" type="checkbox"/>		jtag_uart	JTAG UART		[clk]	
		avalon_jtag_slave	Avalon Memory Mapped Slave	Double-click to export	pll_outclk0	
		irq	Interrupt Sender	Double-click to export	[clk]	
<input checked="" type="checkbox"/>		onchip_memory	On-Chip Memory (RAM or ROM)		[clk 1]	
		s1	Avalon Memory Mapped Slave	Double-click to export	pll_outclk0	
<input checked="" type="checkbox"/>		sram	Generic Tri-State Controller		[clk]	
		uas	Avalon Memory Mapped Slave	Double-click to export	pll_outclk0	
		tcm	Tristate Conduit Master	Double-click to export	[clk]	
<input checked="" type="checkbox"/>		ext_sram_bus	Tri-State Conduit Bridge		[clk]	
		tcs	Tristate Conduit Slave	Double-click to export	pll_outclk0	
		out	Conduit	<b>ext_sram_bus_out</b>		
<input checked="" type="checkbox"/>		epcq	Altera Serial Flash Controller		[clock_sink]	
		avl_csr	Avalon Memory Mapped Slave	Double-click to export	pll_outclk 1	
		avl_mem	Avalon Memory Mapped Slave	Double-click to export	[clock_sink]	
		interrupt_sender	Interrupt Sender	Double-click to export	[clock_sink]	
<input checked="" type="checkbox"/>		led_pio	PIO (Parallel I/O)		[clk]	
		s1	Avalon Memory Mapped Slave	Double-click to export	pll_outclk0	
		external_connection	Conduit	<b>led_pio_external_connection</b>		

## 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 ⇒ 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。

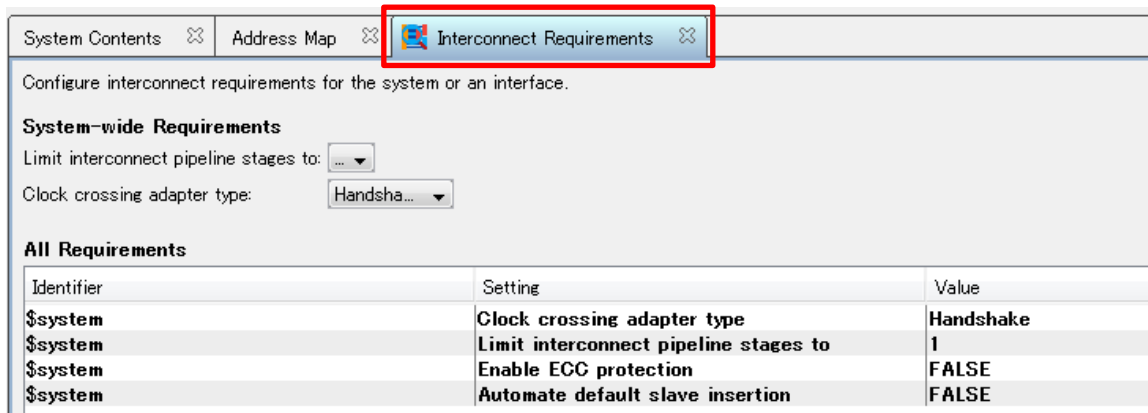


Address Map タブをクリックすると、設定されたベース・アドレスの一覧を確認することができます。

System Contents			Address Map	Interconnect Requirements	
System: nios2_system Path: pll					
		nios2_cpu.data_master		nios2_cpu.instruction_master	
epcqavl_csr		0x0424_1000 - 0x0424_101f			
epcqavl_mem		0x0200_0000 - 0x03ff_ffff		0x0200_0000 - 0x03ff_ffff	
itag_uart.avalon_itag_slave		0x0424_1030 - 0x0424_1037			
led_pios1		0x0424_1020 - 0x0424_102f			
nios2_cpu.debug_mem_slave		0x0424_0800 - 0x0424_0fff		0x0424_0800 - 0x0424_0fff	
onchip_memory.s1		0x0422_0000 - 0x0423_ffff		0x0422_0000 - 0x0423_ffff	
sram.uas		0x0410_0000 - 0x041f_ffff		0x0410_0000 - 0x041f_ffff	

## 5-2. Qsys インタコネクッ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。



### ◆ 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。

### ◆ Clock crossing adapter type

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 : 単純なハンドシェイク・プロトコルを使用します。

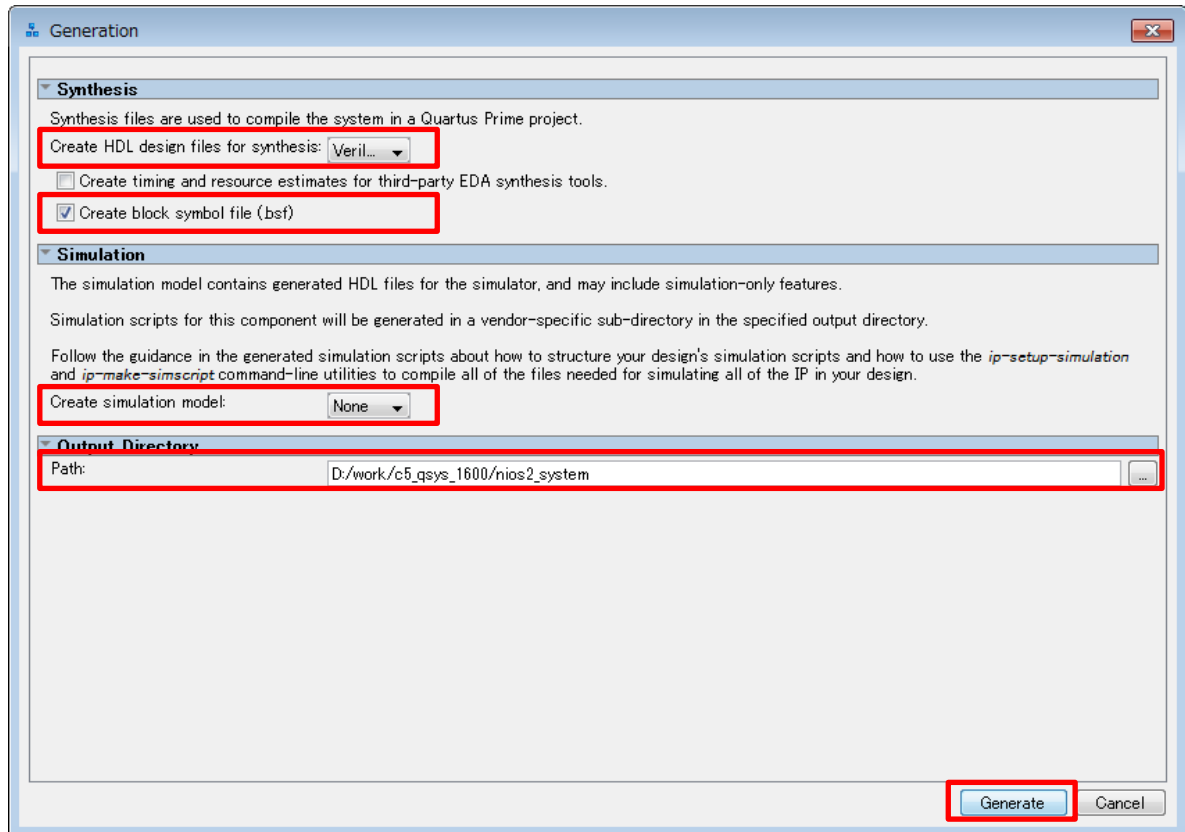
FIFO : 同期のため、デュアル・クロック FIFO を使用します。

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。



#### ◆ Create HDL design file for synthesis

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。

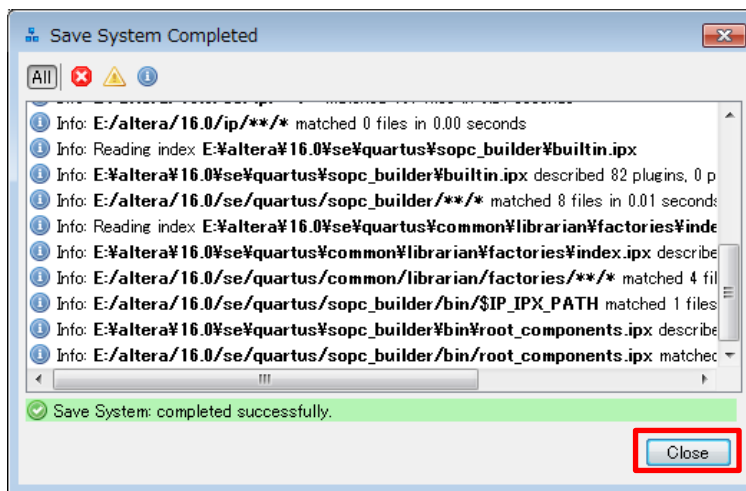
#### ◆ Create block symbol file (.bsf)

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。

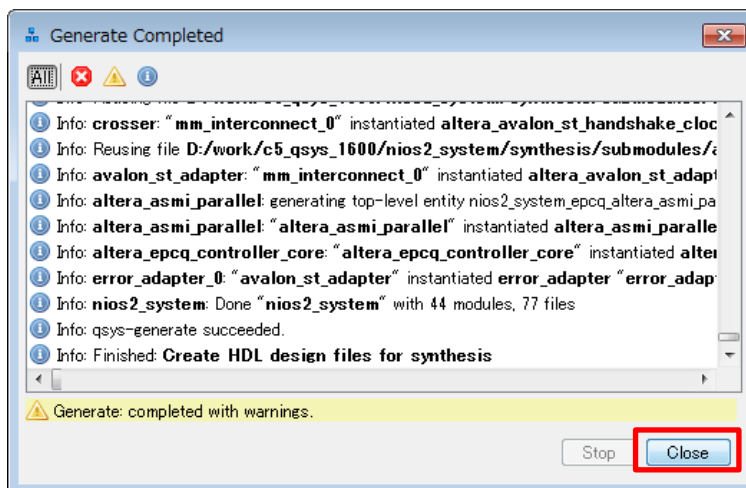
#### ◆ Create simulation model

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。

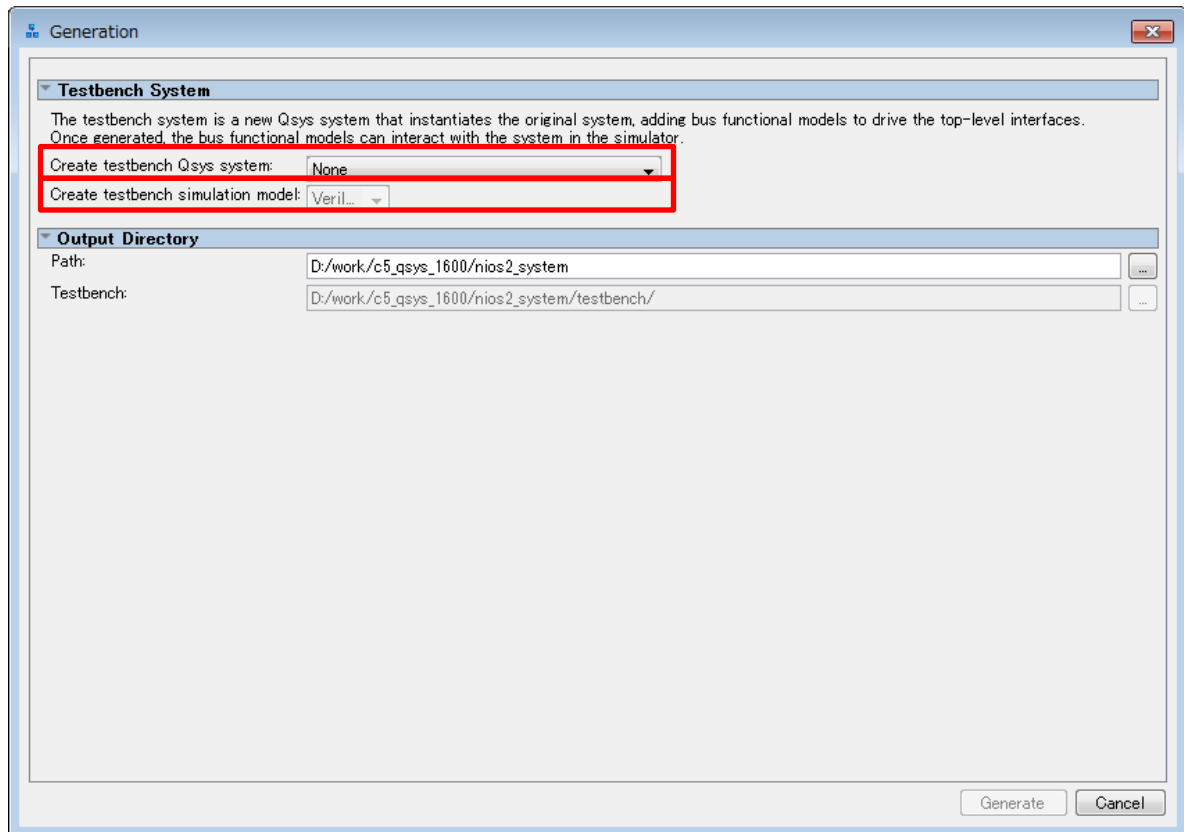


エラーなく生成が終わったら、Close をクリックします。



#### 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 ⇒ 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。



##### ◆ Create testbench Qsys system

- Standard, BFM's 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, BFM's 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。

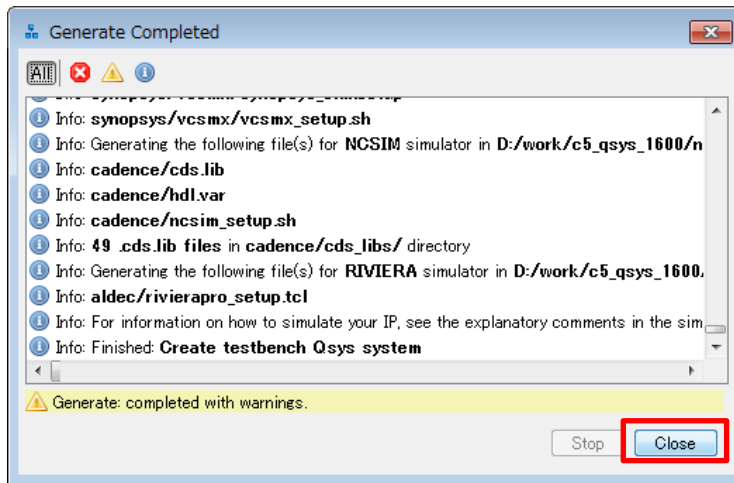
※ BFM:バス・ファンクション・モデル。バスの動作をシミュレーションで確認するためのモデル。

##### ◆ Create testbench simulation model

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。

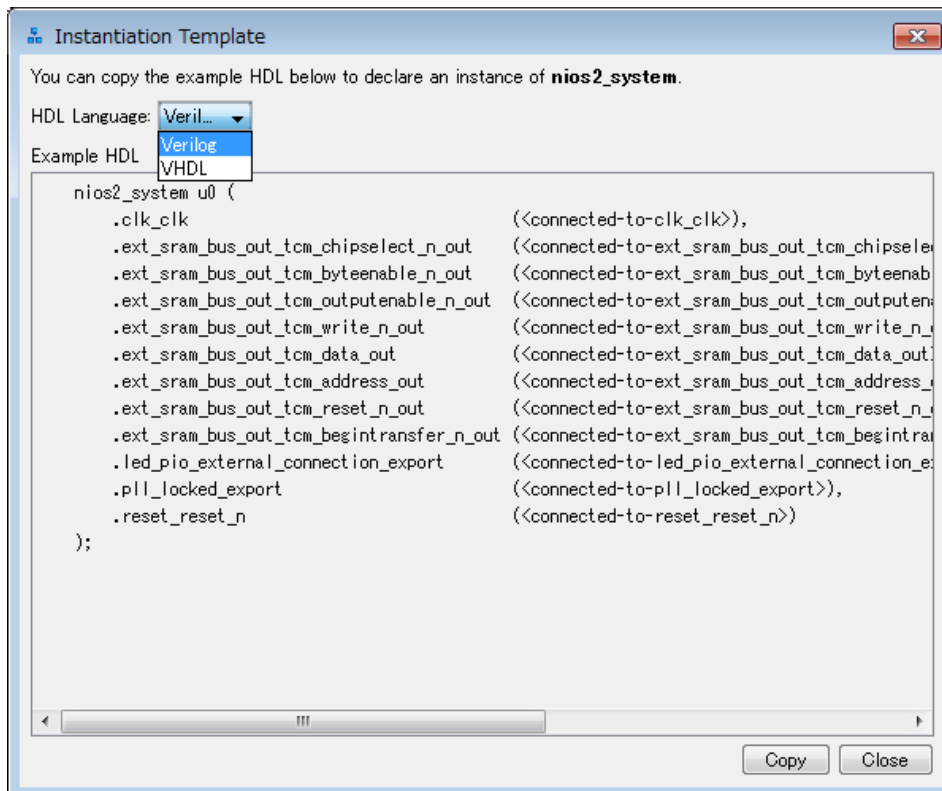


設定が終わったら、Generate をクリックします。そして、エラーなく生成が終わったら、Close をクリックします。



### 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。



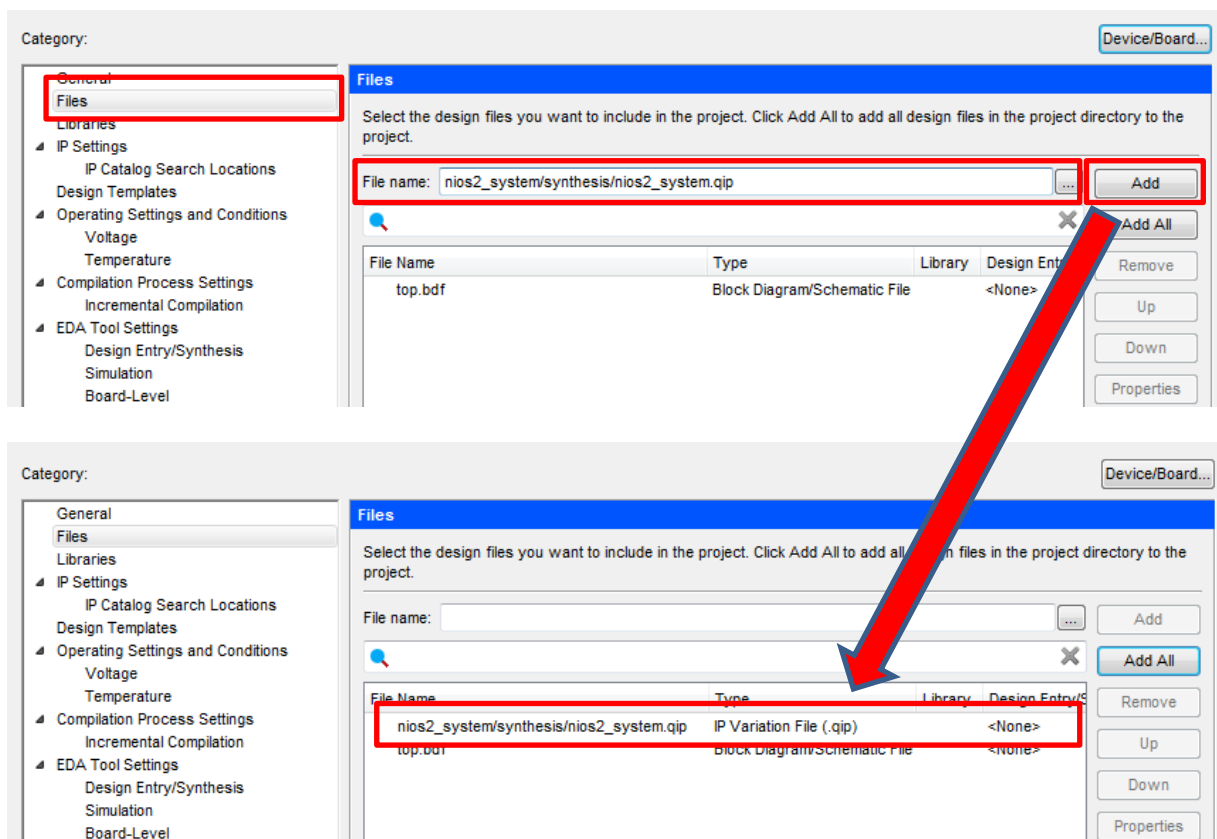
## 6. Quartus Prime での各種設定

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。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



### 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。

## 改版履歴

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。
3. 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: <https://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 : <https://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、各メーカー発行の英語版の資料もあわせてご利用ください。