

Quartus® Prime はじめてガイド  
Quartus® Prime Standard Edition  
開発ソフトウェア  
簡易チュートリアル

# Quartus® Prime Standard Edition 開発ソフトウェア 簡易チュートリアル

## 目次

本書をお読みにする前に.....	3
はじめに .....	4
チュートリアルを実施するために必要な環境.....	4
サンプルデザインの概要.....	5
Quartus® Prime での開発 .....	6
1. Quartus® Prime を起動する.....	6
2. 論理回路を設計する.....	7
2-1. プロジェクトを作成する .....	7
2-2. 論理回路を作成する.....	11
2-3. Analysis and Elaboration を実行する .....	25
3. 論理シミュレーションを実行する .....	27
4. 制約の設定をする.....	33
4-1. デザイン上のピンをデバイスにアサインする.....	33
4-2. タイミング制約.....	35
5. コンパイルを実行する.....	36
6. タイミングを検証する .....	38
7. プログラミングをする.....	39
7-1. sof ファイルを書き込む .....	40
7-2. コンフィグレーション ROM へ書き込む .....	43
7-2-1. コンフィグレーション ROM 用のプログラミング・ファイルを作成する.....	43
7-2-2. コンフィグレーション ROM へプログラミングする.....	47
8. 実機で動作検証をする.....	52

## 本書をお読みにする前に

この資料の内容は 2024 年 10 月現在のものです。

この資料で紹介しているソフトウェアやハードウェア、操作手順などは、指定バージョンやデバイスなど以外でも共通のものもありますが、一部については共通にならないものもありますので、ご注意ください。

### 文書中の記号

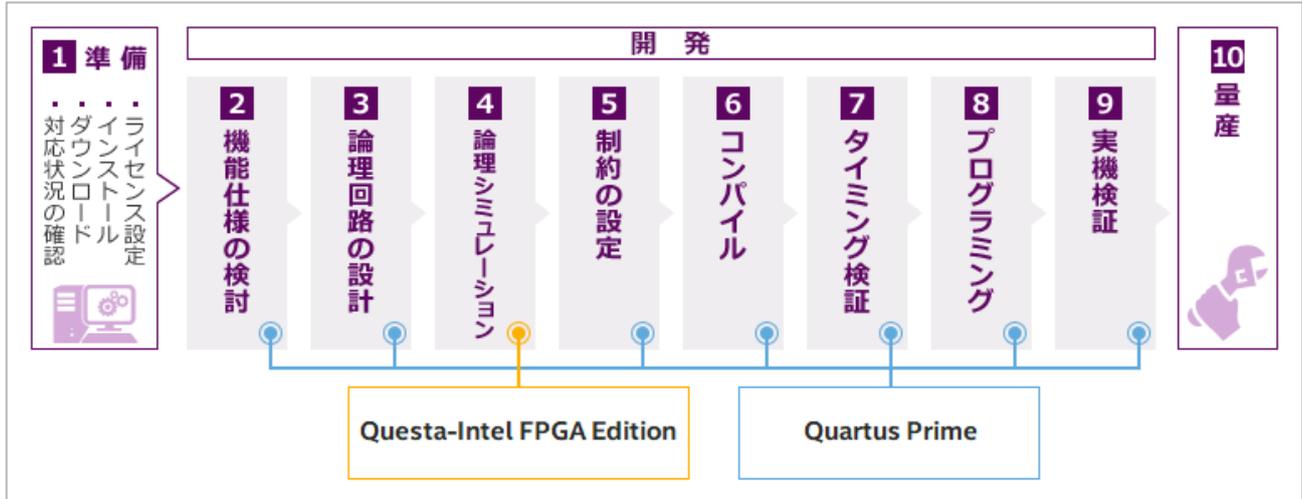
① Note	補足情報などを記載しています。
② Point	重要なポイントを記載しています。
📖 参考	理解を深めるため、参考となる資料やサイトを紹介しています。
⚠️ 注記	この資料の中では具体的には触れませんが、必要となる知識や情報を記載しています。
🚫 禁止	注意点や、してはいけないことを記載しています。

### 文中の表記

<u>下線</u>	クリックする事で、資料中の別の章や、外部のサイトにジャンプします。
<b>太字斜体</b>	画面の操作をする際の、メニューやウィンドウなどに表示されている文字を示しています。
xxxxxxx	入力するコマンド文字列を示しています。

## はじめに

この資料は、Quartus® Prime Standard Edition 開発ソフトウェア (以降 Quartus® Prime) を使用して、デジタル論理回路の作成からインテル FPGA の搭載された評価ボードを動作させるまでをチュートリアル形式にまとめたものです。はじめて FPGA 設計をするユーザーが、開発ソフトウェアの基本的な操作フローを簡易的に理解することができます。



上図は、FPGA の一般的な開発フローを示しています。

インテル FPGA を開発する場合、開発フローの [2]~[9] は Quartus® Prime を使用し、そのうち開発フローの [4] では Quarta\* - Intel® FPGA Edition などのインテル FPGA をサポートする EDA シミュレーション・ツールを使用します。

このチュートリアルでは、開発フローの [3]~[9] を体験することができます。

なお、各工程の操作マニュアルは、下記ページよりご利用いただけます。

[Altera® FPGA の開発フロー](#)

## チュートリアルを実施するために必要な環境

このチュートリアルを実施するためには、以下の環境が必要です。

【表 A】チュートリアル実施に必要な環境

No.	項目	内容
1	パソコン	Quartus® Prime が動作するスペックを保有するパソコンをご用意ください。 ・ サポート OS 情報は、 <a href="#">Quartus® Prime 開発ソフトウェア - サポート OS 対応表</a> をご覧ください。 ・ パソコンの物理メモリ容量の情報は、 <a href="#">こちらのページ</a> から 使用するバージョンの Release Notes を入手し、Disk Space and Memory Recommendations のセクションをご覧ください。

No.	項目	内容									
2	Quartus® Prime Standard Edition	開発ソフトウェアをインストールしてください。各種ツールのダウンロードおよびインストール方法は、以下の Web ページをご参照ください。									
3	Questa*- Intel® FPGA Edition または Questa*- Intel® FPGA Starter Edition	<ul style="list-style-type: none"> <li>・ <a href="#">Quartus® Prime 開発ソフトウェアおよび Questa* - Intel® FPGA Edition のダウンロード方法</a></li> <li>・ <a href="#">Quartus® Prime 開発ソフトウェアおよび Questa* - Intel® FPGA Edition のインストール方法</a></li> </ul>									
4	評価ボード	<p>このチュートリアルに対応した評価ボードは、以下のとおりです。</p> <table border="1"> <thead> <tr> <th>ボード</th> <th>FPGA</th> <th>Altera® FPGA ダウンロード・ケーブル の有無</th> </tr> </thead> <tbody> <tr> <td><a href="#">Cyclone® 10 LP Evaluation Kit</a></td> <td>10CL025YU256I7G</td> <td>不要 (別途使用可能)</td> </tr> <tr> <td><a href="#">MAX® 10 FPGA Evaluation Kit</a></td> <td>10M08SAE144C8G</td> <td>必要</td> </tr> </tbody> </table>	ボード	FPGA	Altera® FPGA ダウンロード・ケーブル の有無	<a href="#">Cyclone® 10 LP Evaluation Kit</a>	10CL025YU256I7G	不要 (別途使用可能)	<a href="#">MAX® 10 FPGA Evaluation Kit</a>	10M08SAE144C8G	必要
ボード	FPGA	Altera® FPGA ダウンロード・ケーブル の有無									
<a href="#">Cyclone® 10 LP Evaluation Kit</a>	10CL025YU256I7G	不要 (別途使用可能)									
<a href="#">MAX® 10 FPGA Evaluation Kit</a>	10M08SAE144C8G	必要									
5	Altera® FPGA ダウンロード・ケーブル II (旧称 USB-Blaster II) または Altera® FPGA ダウンロード・ケーブル (旧称 USB-Blaster)	<p>開発したデータを FPGA やコンフィグレーションデバイスにダウンロードするための専用 USB ケーブルです。</p> <ul style="list-style-type: none"> <li>・ 評価ボード Cyclone® 10 LP Evaluation Kit を使用する場合は、これらのダウンロード・ケーブルは必要ありません。Altera® FPGA ダウンロード・ケーブル II と同等の機能を基板に搭載しています。</li> <li>・ これらダウンロード・ケーブルをはじめて利用するパソコンには、事前にドライバをインストールしてください。インストール方法は、以下の Web ページをご参照ください。 <a href="#">USB-Blaster™ II のドライバをインストールしてみよう</a> <a href="#">USB-Blaster™ のドライバをインストールしてみよう</a></li> </ul>									

## サンプルデザインの概要

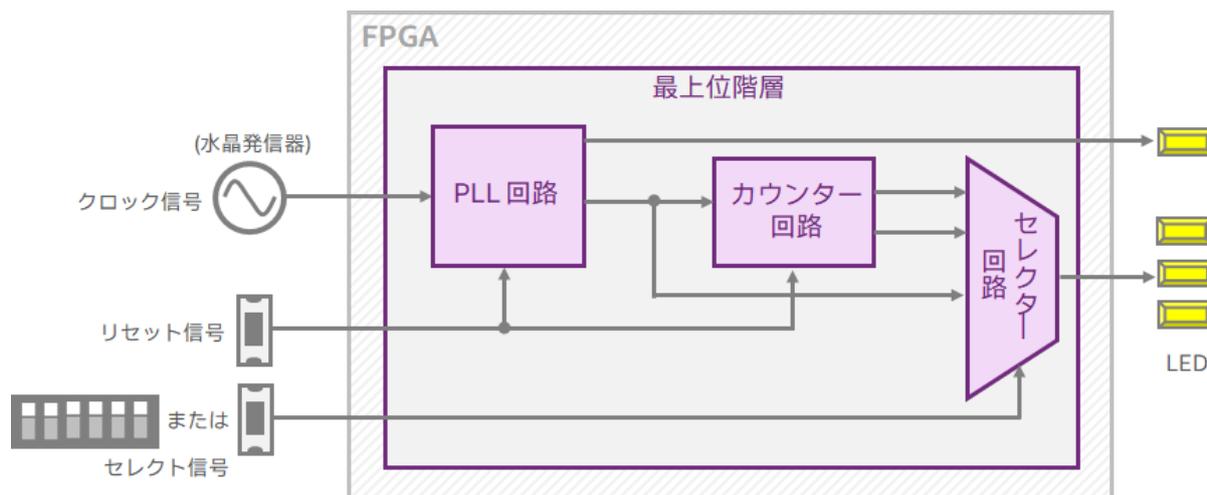
このチュートリアルでは、評価ボード上の LED の点灯・消灯を制御する (いわゆる L チカ) 動作をデジタル論理回路としてハードウェア言語 (以下 HDL) で設計します。

作成するデザインは、3つのブロック (機能) で構成します。

### ◆ 各ブロックの概要

デザイン	動作概要
PLL (Phase-Locked Loop) 回路	外部から入力されたクロック信号の周波数を逡倍・分周します
カウンター回路	非同期クリアー付き 32bit アップカウンター
セレクター回路 (マルチプレクサー)	入力条件に応じて出力するデータを選択します

サンプル設計のブロックイメージは、次のとおりです。



#### 参考:

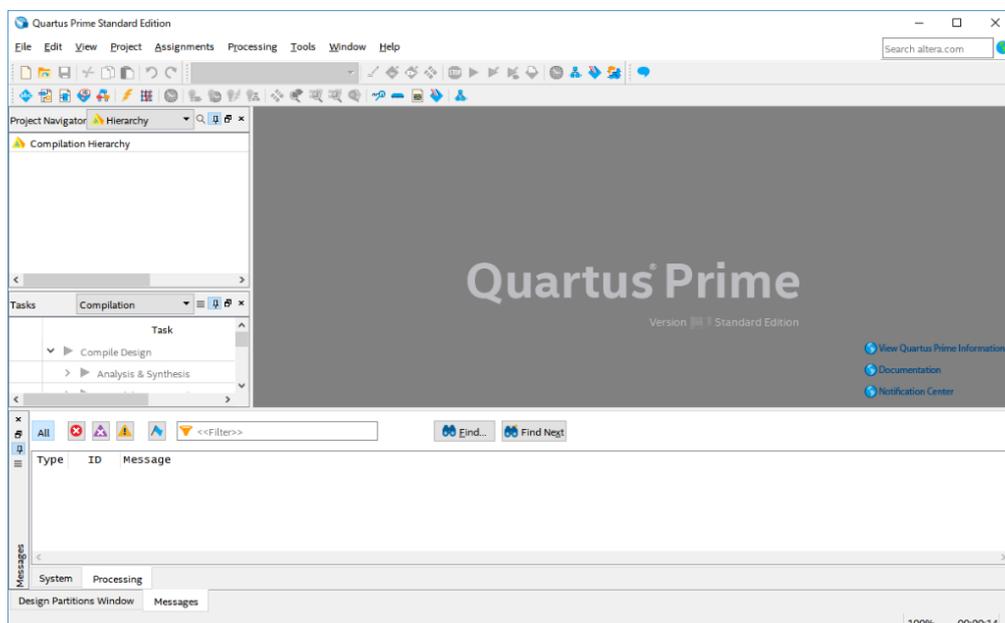
- HDL 設計経験がないユーザーは、以下もご活用ください。
  - [はじめてみよう！ Verilog-HDL<演習問題つき>](#)
  - [はじめてみよう！ VHDL<演習問題つき>](#)

## Quartus® Prime での開発

### 1. Quartus® Prime Standard Edition を起動する

Windows OS の場合は、**[スタート] ▶ Intel FPGA <version\_build> Standard Edition ▶ Quartus (Quartus Prime <version>)** をクリック、もしくはデスクトップに生成した Quartus® Prime のショートカット・アイコンをダブルクリックしてください。

Linux OS の場合は、**quartus** コマンドにより起動させます。



## 2. 論理回路を設計する

1つの FPGA に構築する論理回路を作成します。(この作業は、開発フローの [3] にあたります。)

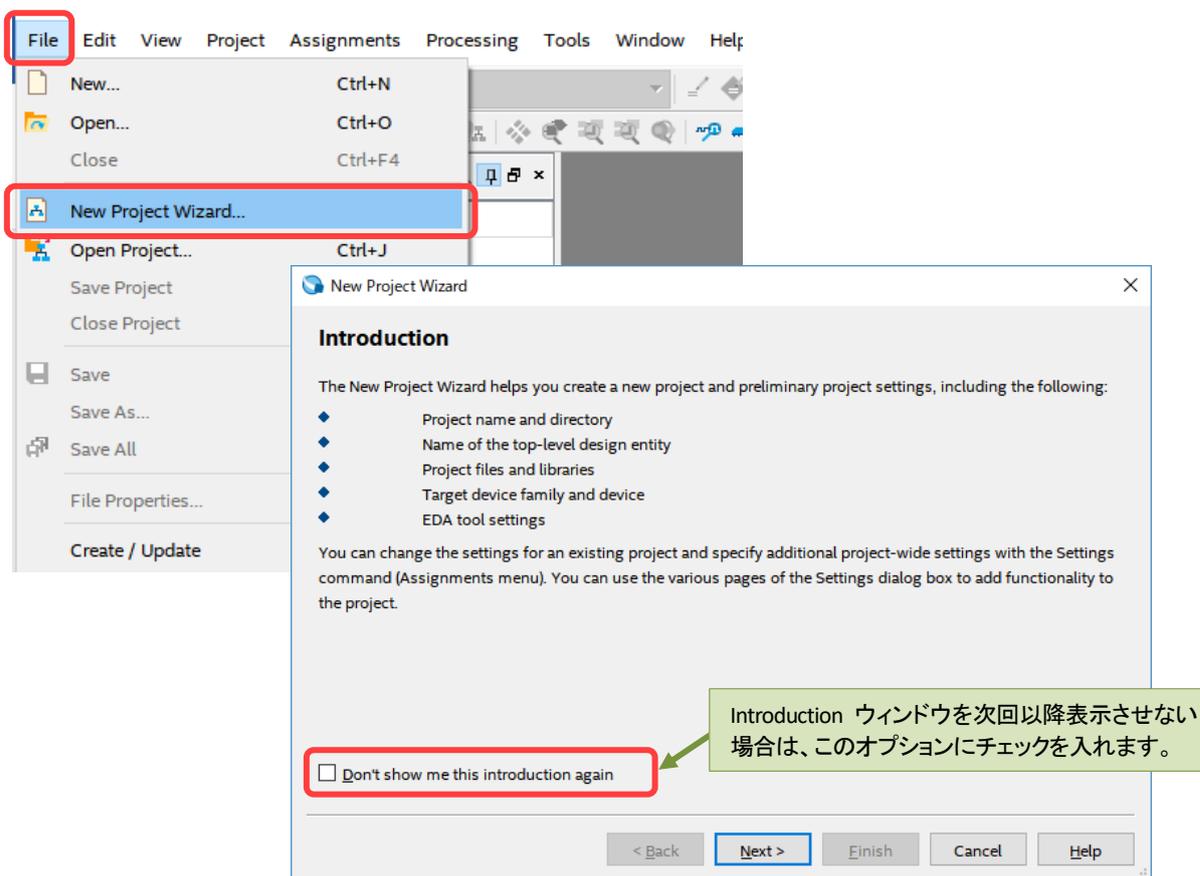


1つの FPGA に対して開発を始めるには、Quartus® Prime Standard Edition で1つのプロジェクトを構築する必要があります。そのプロジェクトに対して、HDL やスケマティックで論理回路を作成し、登録を行います。

### 2-1. プロジェクトを作成する

論理回路を作成するためのプロジェクトを作成します。

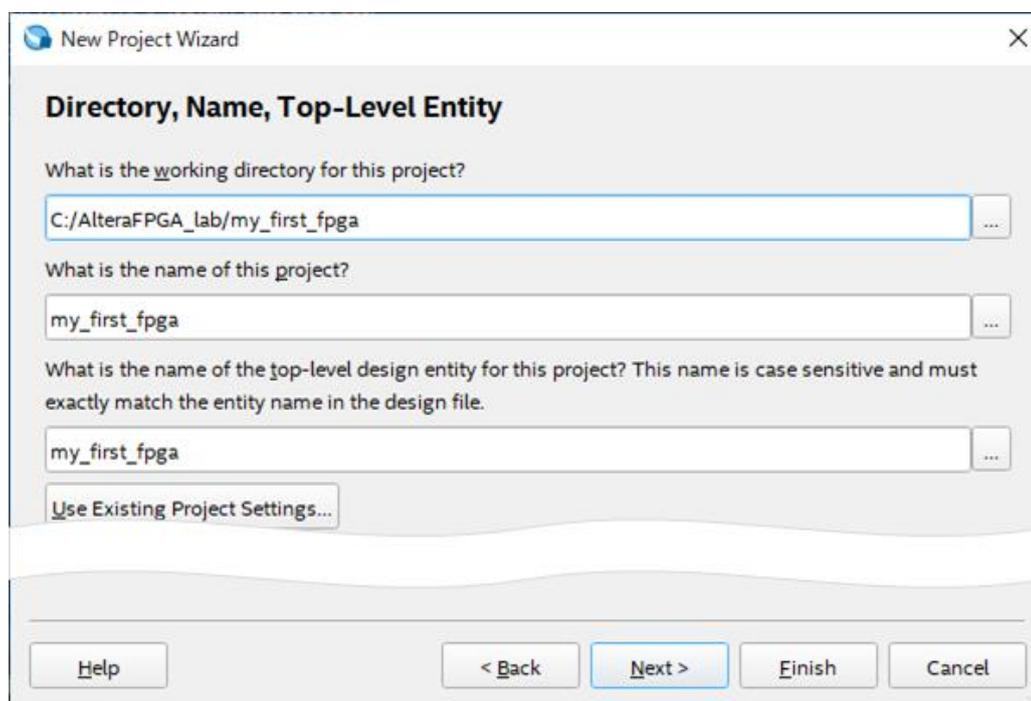
1. Quartus® Prime のメニューから **File > New Project Wizard** を選択します。New Project Wizard ダイアログボックスが起動し、初回は Introduction ページが表示されますので、**Next** ボタンをクリックし、次へ進みます。



2. Directory, Name, Top-Level Entity ページでは、以下の情報を登録し、**Next** ボタンをクリックします。

【表 2-1-1】 Directory, Name, Top-Level Entity ページに入力する内容

質問事項	入力内容
<i>What is the working directory for this project?</i>	プロジェクト用 作業フォルダーのパス C:/AlteraFPGA_lab/my_first_fpga
<i>What is the name of this project?</i>	プロジェクト名 my_first_fpga
<i>What is the name of the top-level design entity for this project? This name is ... (省略)</i>	プロジェクトにおける最上位エンティティ名 my_first_fpga



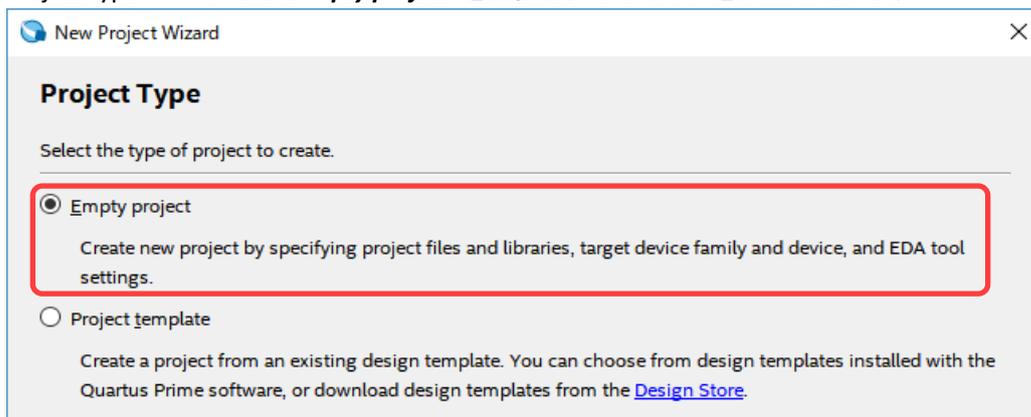
① Note:

作業フォルダーのパス、プロジェクト名、ファイル名は、半角・英数字のみを利用してください。全角やスペースの使用はできません。

このとき、プロジェクト用作業フォルダーをあらかじめ作成していない場合は、指定したフォルダーの自動作成を認証するメッセージ (下図) が表示されます。**Yes** を選択し、作業フォルダーを作成します。



3. Project Type ページでは、**Empty project** を選択し、**Next** ボタンをクリックします。

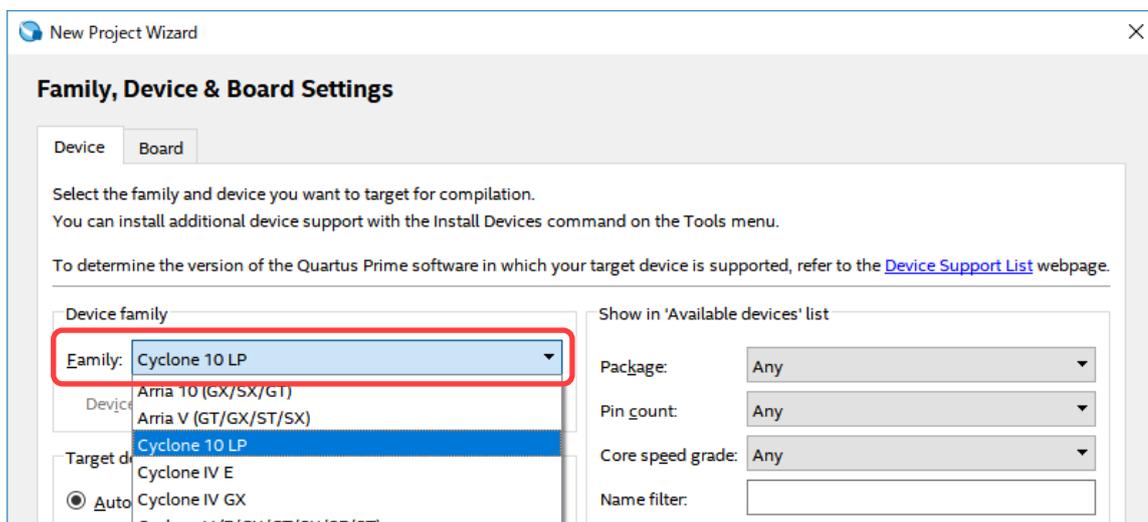


4. Add Files ページでは、既存のデザインファイルを指定します。  
このチュートリアルでは、まだデザインを作成していませんので空欄のまま **Next** ボタンをクリックします。
5. Family, Device & Board Settings ページでは、使用する評価ボードに搭載されている FPGA の型番を選択します。

【表 2-1-2】 Family, Device & Board Settings ページで選択する内容

評価ボード	Device Family	Package	Pin count	Core speed grade	Available devices
Cyclone® 10 LP Evaluation Kit	Cyclone 10 LP	UFBGA	256	7	10CL025YU256I7G
MAX® 10 FPGA Evaluation Kit	MAX 10	EQFP	144	8	10M08SAE144C8G

- ① **Family** (Device family エリア内) プルダウン・リストから、FPGA の デバイスファミリー を選択します。



▲ 例: Cyclone® 10 LP Evaluation Kit をターゲットにした場合

- ② **Show in 'Available devices' list** エリアにおいて、[表 2-1-2] を参考に **Package**、**Pin count**、**Core speed grade** の項目を選択します。

選択した項目がフィルター条件となり、Available devices 欄に該当する型番のみが表示されます。

New Project Wizard

### Family, Device & Board Settings

Device Board

Select the family and device you want to target for compilation.  
You can install additional device support with the Install Devices command on the Tools menu.

To determine the version of the Quartus Prime software in which your target device is supported, refer to the [Device Support List](#) webpage.

Device family

Family: Cyclone 10 LP  
Device: All

Target device

Auto device selected by the Fitter  
 Specific device selected in 'Available devices' list  
 Other: n/a

Show in 'Available devices' list

Package: UFBGA  
Pin count: 256  
Core speed grade: 7  
Name filter: 025  
 Show advanced devices

▲ 例: Cyclone® 10 LP Evaluation Kit をターゲットにした場合

④ Point:

条件を入力しても **Available devices** 覧に多くの型番が検出され対象の FPGA が見つけにくいときは、**Show in 'Available devices' list** エリアの **Name filter** 欄にキーワードを入力して、さらに絞り込んでください。

例えば、Cyclone 10 LP Evaluation Kit (10CL025YU256I7G) の場合、Name filter 欄に “025” を入力するとよいでしょう。

- ③ **Available devices** 欄から対象の型番をクリックで指定 (ハイライト) し、**Finish** ボタンをクリックします。

Target device

Auto device selected by the Fitter  
 Specific device selected in 'Available devices' list  
 Other: n/a

Core speed grade: 7  
Name filter: 025  
 Show advanced devices

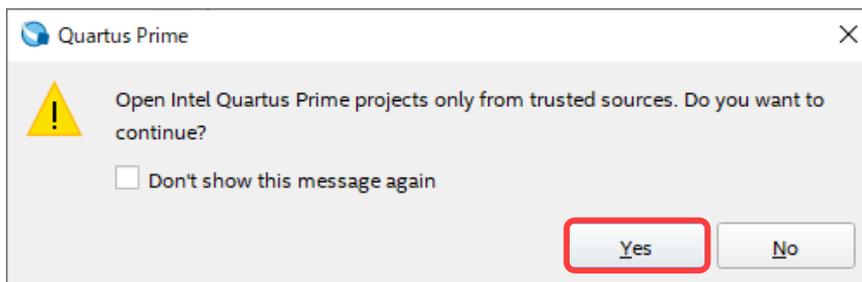
Available devices:

Name	Core Voltage	LEs	Total I/Os	GPIOs	Memory Bits	Embedded multiplier 9-bit
10CL025YU256A7G	1.2V	24624	151	151	608256	132
10CL025YU256I7G	1.2V	24624	151	151	608256	132

< Back Next > Finish Cancel Help

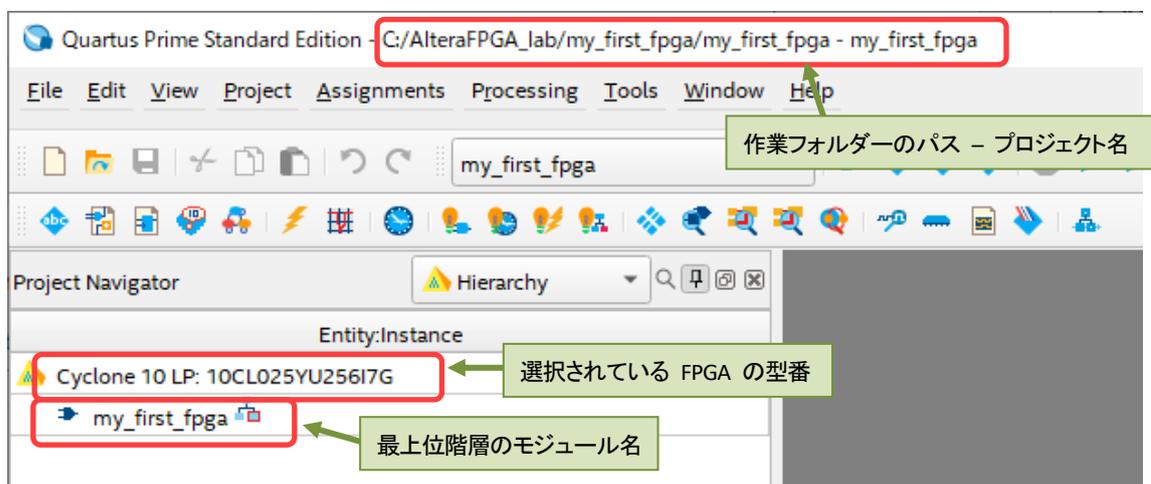
▲ 例: Cyclone® 10 LP Evaluation Kit をターゲットにした場合

6. 以下のメッセージが表示されます。作業を継続するため **Yes** ボタンをクリックします。



7. 以上で プロジェクト作成が完了しました。

画面のツールバー、および Project Navigator ウィンドウに、設定した情報が表示されていることを確認します。



▲ 例: Cyclone® 10 LP Evaluation Kit をターゲットにした場合

#### 参考:

- [「Quartus Prime はじめてガイド - プロジェクトの作成方法」](#)

## 2-2. 論理回路を作成する

作成したプロジェクトに、[サンプルデザインの概要](#) で示した論理回路を作成します。

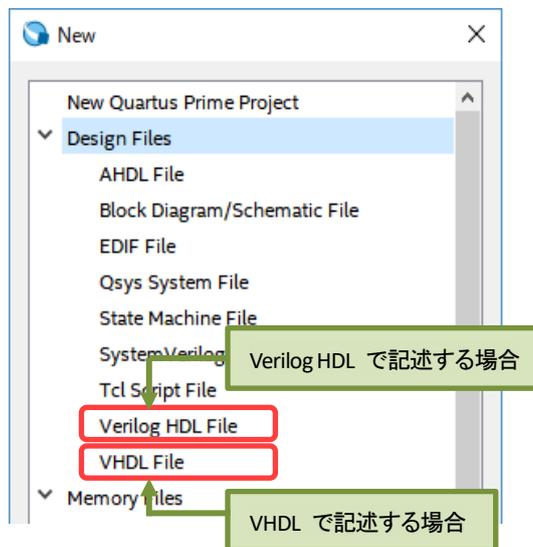
このチュートリアルでは、カウンター回路 および セレクター回路は HDL (VHDL または Verilog HDL) でコードを記述し、PLL 回路は IP Catalog という入力手法で設計します。また、この3つのエンティティーを接続する最上位階層デザインも、HDL で記述します。

■ カウンター回路を設計する

VHDL または Verilog HDL で記述します。

1. Quartus® Prime のメニューから **File ▶ New** を選択します。表示された New ダイアログボックスの Design Files ブランチ内から、記述する HDL 用のファイルを選択し、OK ボタンをクリックします。

新規ワーキングシートが開きます。

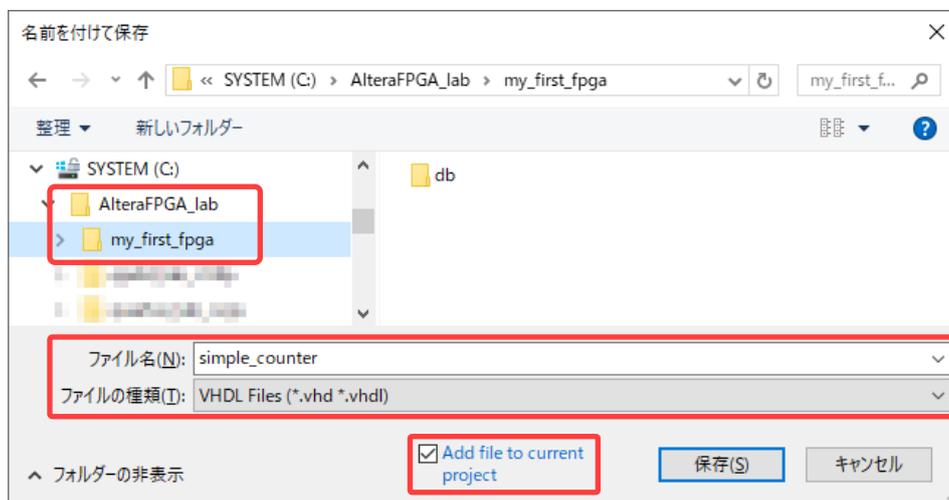


- 2.
3. Quartus® Prime のメニューから **File ▶ Save As ...** を選択します。

これから作成するデザインファイルを保存するフォルダーを指定し、デザインファイルに名前を付け、**保存(S)** ボタンをクリックします。

【表 2-1-3】 デザインファイルを保存する際の指定内容

項目	指定内容
保存するフォルダー	C:/AlteraFPGA_lab/my_first_fpga
ファイル名	simple_counter
ファイルの種類	VHDL で記述 ⇒ VHDL Files (*.vhd *.vhdI)
	Verilog HDL で記述 ⇒ Verilog HDL Files (*.v *.vlg *.verilog)
Add file to current project オプション	On



▲ 例: VHDL で記述した場合の保存画面

4. 先ほど保存したワーキングシート (simple\_counter.vhd または simple\_counter.v) に HDL コードを入力します。

下記に各 HDL の記述サンプルを掲載していますので、参考にしてください。

【サンプル 2-1-1】 simple\_counter.vhd コード例

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity simple_counter is
  port
  (
    clock      : in std_logic;
    reset      : in std_logic;
    counter_out : out std_logic_vector (31 downto 0)
  );
end entity;

architecture rtl of simple_counter is
  signal cnt : std_logic_vector (31 downto 0) := (others => '0');

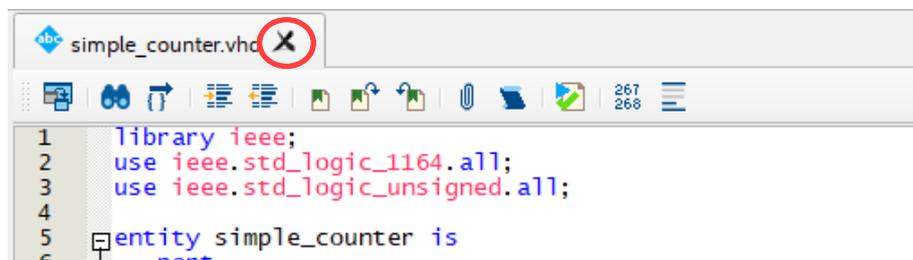
begin
  process (clock, reset)
  begin
    if (reset = '0') then
      cnt <= (others => '0');
    elsif (rising_edge(clock)) then
      cnt <= cnt + 1;
    end if;
    counter_out <= cnt;
  end process;
end rtl;
```

【サンプル 2-1-2】 simple\_counter.v コード例

```
module simple_counter
(
  input clock,
  input reset,
  output reg [31:0] counter_out
);

always @ (posedge clock or negedge reset)
begin
  if (!reset)
    counter_out <= 0;
  else
    counter_out <= counter_out + 1;
  end
endmodule
```

5. Quartus® Prime のメニューから **File ▶ Save** を選択し、デザインファイルを保存します。
6. ワーキングシートのタブ右端にある [X] ボタンをクリックし、ファイルを閉じます。



以上で カウンター回路の設計は終了です。

## ■ セレクター回路を設計する

VHDL または Verilog HDL で記述します。

1. Quartus® Prime のメニューから **File ▶ New** を選択します。表示された New ダイアログボックスの Design Files ブランチ内から、記述する HDL 用のファイルを選択し、**OK** ボタンをクリックします。  
新規ワーキングシートが開きます。
2. Quartus® Prime のメニューから **File ▶ Save As ...** を選択します。  
これから作成するデザインファイルを保存するフォルダーを指定し、デザインファイルに名前を付け、**保存(S)** ボタンをクリックします。

【表 2-1-4】 デザインファイルを保存する際の指定内容

項目	指定内容
保存するフォルダー	C:/AlteraFPGA_lab/my_first_fpga
ファイル名	counter_bus_mux
ファイルの種類	VHDL で記述 ⇒ VHDL Files (*.vhd *.vhd1)
	Verilog HDL で記述 ⇒ Verilog HDL Files (*.v *.vlg *.verilog)
Add file to current project オプション	On

3. 先ほど保存したワーキングシート (counter\_bus\_mux.vhd または counter\_bus\_mux.v) に HDL コードを入力します。  
次頁に各 HDL の記述サンプルを掲載していますので、参考にしてください。

【サンプル 2-1-3】 counter\_bus\_mux.vhd コード例

```
library ieee;
use ieee.std_logic_1164.all;

entity counter_bus_mux is
    port
    (
        dataa    : in std_logic_vector (2 downto 0);
        datab   : in std_logic_vector (2 downto 0);
        sel      : in std_logic;
        result   : out std_logic_vector (2 downto 0)
    );
end entity;
architecture rtl of counter_bus_mux is
begin

    result <= dataa when (sel = '1') else datab;

end rtl;
```

【サンプル 2-1-4】 counter\_bus\_mux.v コード例

```
module counter_bus_mux
(
    input  [2:0] dataa,
    input  [2:0] datab,
    input          sel,
    output [2:0] result
);

    assign result = (sel == 1'b1) ? dataa : datab;

endmodule
```

4. Quartus® Prime のメニューから **File ▶ Save** を選択し、デザインファイルを保存します。
5. ワーキングシートのタブ右端にある [X] ボタンをクリックし、ファイルを閉じます。

以上で セレクター回路の設計は終了です。

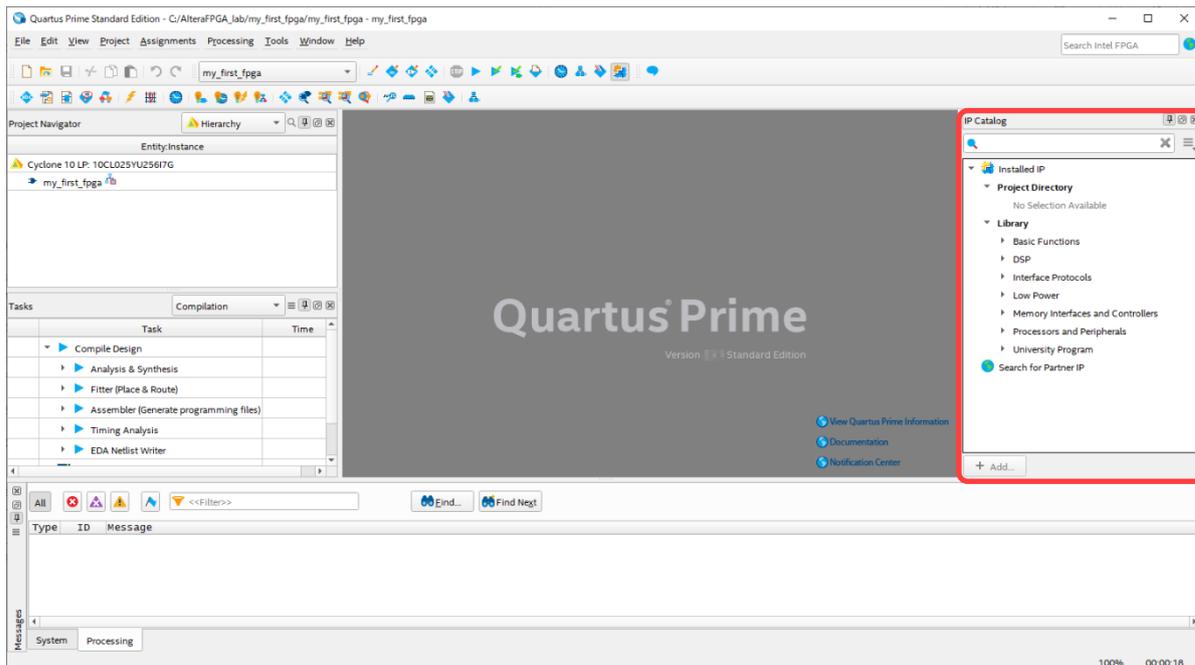
■ PLL 回路を設計する

IP Catalog で作成します。

① Note:

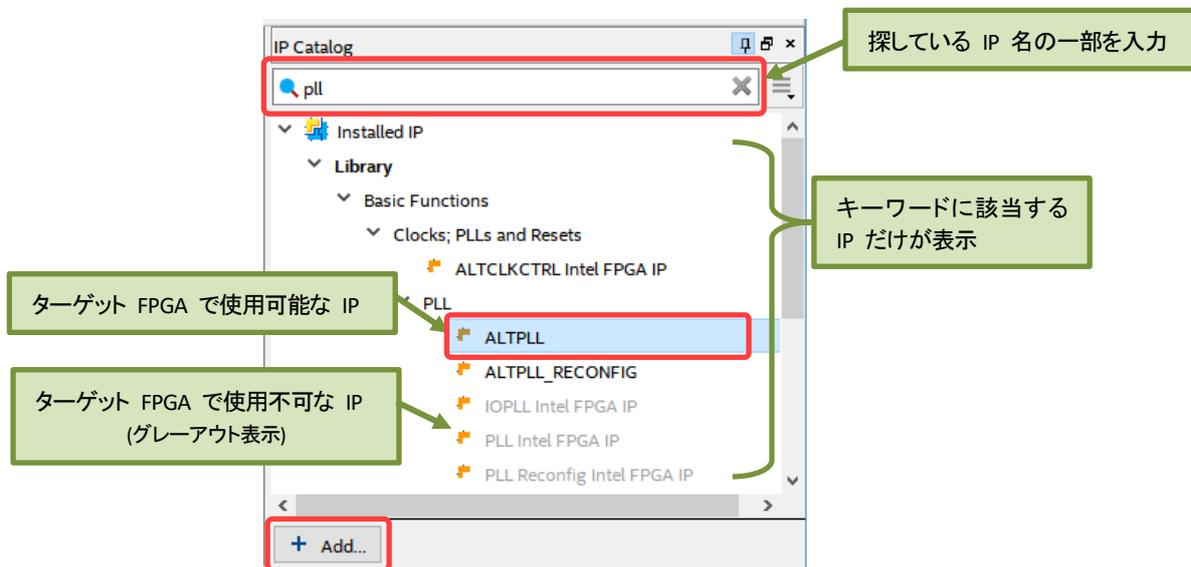
IP Catalog とは、インテル FPGA の IP コアのパラメーターを簡単にカスタマイズして、プロジェクトに組み込むことができるツールです。詳細は、[Introduction to Intel FPGA IP Cores](#) をご参考ください。

1. Quartus® Prime の画面右端に IP Catalog ウィンドウが表示されていない場合は、メニューから **Tools > IP Catalog** を選択し、IP Catalog ウィンドウを表示します。



2. IP Catalog の検索フィールドに PLL IP 名の一部である *pll* と入力すると、指定したキーワードを IP 名に持つラインナップのみが表示されます。

**ALTPLL** を選択し、**+ Add** ボタンをクリックします。



## ① Note:

デバイスファミリーにより指定する PLL の IP 名が異なります。詳細は下記ページをご参考ください。

- [IP の生成 \(「インテル® FPGA の PLL」より\)](#)

## 3. Save IP Variation ダイアログボックスが表示されます。

これから作成する PLL IP の保存フォルダー、およびエンティティー名を入力し、生成する HDL フォーマットを選択します。

このチュートリアルでは、以下のとおりに設定します。

【表 2-1-5】 デザインファイルを保存する際の指定内容

項目	内容
IP variation file name	C:/AlteraFPGA_lab/my_first_fpga/pll
IP variation file type	VHDL で記述 ⇒ VHDL を選択
	Verilog HDL で記述 ⇒ Verilog HDL を選択



▲ 例: VHDL を選択した場合の画面

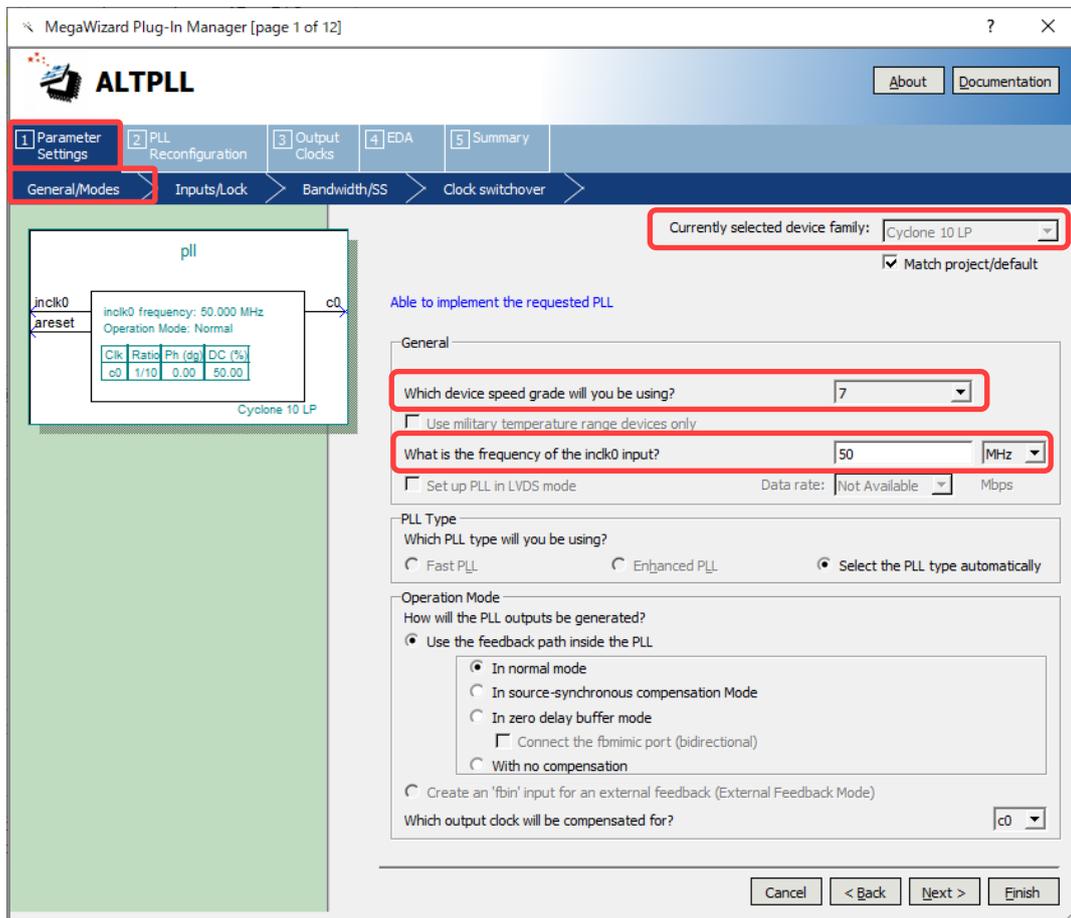
OK ボタンをクリックすると、Wizard 画面が起動します。

4. ALTPLL IP のパラメーターを設定します。

- ① MegaWizard Plug-In Manager ウィンドウの **1** *Parameter Settings* > *General/Modes* において、[表 2-1-6] を参考にパラメーターを設定します。

【表 2-1-6】 ALTPLL のパラメーター設定 (General/Modes)

項目	Cyclone® 10 LP Evaluation Kit	MAX®10 FPGA Evaluation Kit
<i>Currently selected device family</i>	Cyclone 10 LP	MAX 10
<i>Which device speed grade will you be using?</i>	7	8
<i>What is the frequency of the inclock0 input?</i>	50 (MHz)	50 (MHz)
<i>その他の設定</i>	デフォルト	デフォルト



▲ 例: Cyclone® 10 LP Evaluation Kit をターゲットにした場合

**Next** ボタンをクリックします。

② 1 Parameter Settings > Inputs/Lock において、[表 2-1-7] を参考にパラメーターを設定します。

【表 2-1-7】 ATPLL のパラメーター設定 (Inputs/Lock)

項目	Cyclone® 10 LP Evaluation Kit	MAX® 10 FPGA Evaluation Kit
Create an 'areset' input to asynchronously reset the PLL	On	On
Create 'locked' output	On	On
その他の設定	Off	Off

The screenshot shows the ATPLL configuration window in the MegaWizard Plug-In Manager. The 'Inputs/Lock' tab is active. In the 'Optional Inputs' section, the checkbox for 'Create an 'areset' input to asynchronously reset the PLL' is checked. In the 'Lock Output' section, the checkbox for 'Create 'locked' output' is checked. A callout box with a green border and white background points to the 'areset' and 'locked' ports in the graphical preview, containing the text: 'オプションポートは、選択されたポートのみがグラフィカルプレビューに表示'.

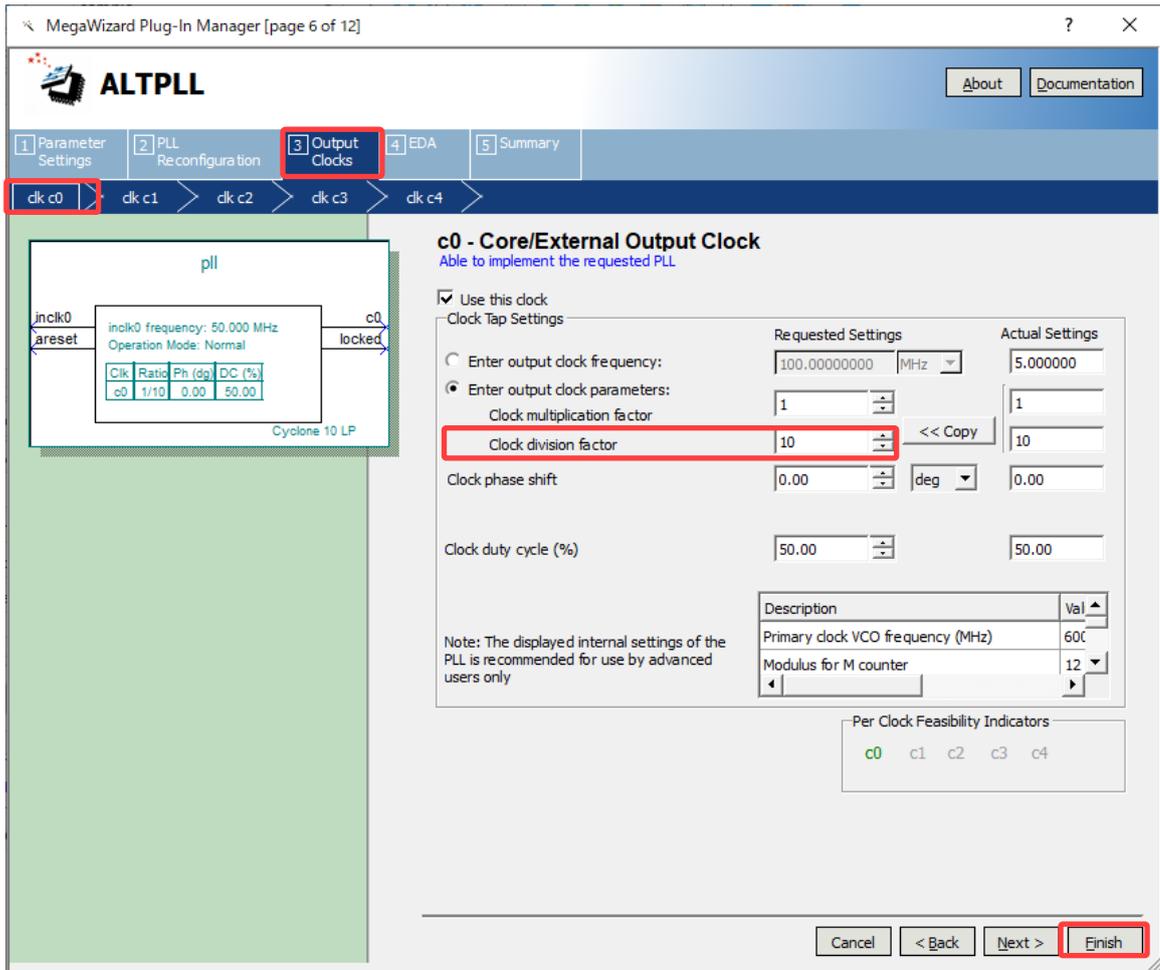
▲ 例: Cyclone® 10 LP Evaluation Kit をターゲットにした場合

- ③ **3** *Output Clocks* をクリックし、*clk c0* ページへジャンプします。

[表 2-1-8] を参考にパラメーターを設定します。

【表 2-1-8】ATLPLL のパラメーター設定 (clk c0)

項目	Cyclone® 10 LP Evaluation Kit	MAX® 10 FPGA Evaluation Kit
<i>Clock division factor</i>	10	10
<i>その他の設定</i>	デフォルト	デフォルト



▲ 例: Cyclone® 10 LP Evaluation Kit をターゲットにした場合

- ④ *Finish* ボタンをクリックします。

- ⑤ **5** *Summary* が表示されます。ここでは、IP ファイル以外に追加生成するファイルを選択します。

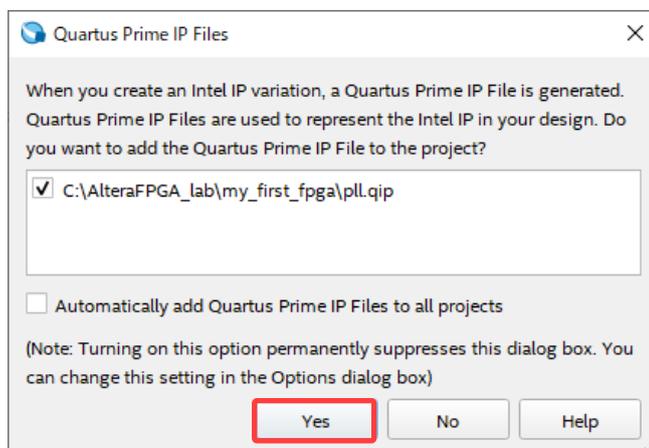
このチュートリアルでは、[表 2-1-9] を参考に選択してください。

【表 2-1-9】ATLPLL のパラメーター設定 (Summary)

言語	選択するファイル
VHDL で設計	pll.cmp、pll_inst.vhd
Verilog HDL で設計	pll_inst.v

⑥ **Finish** ボタンをクリックします。

以下のダイアログボックスが表示されますので、**Yes** ボタンをクリックします。



以上で PLL 回路の設計は終了です。

■ **最上位階層デザイン**を設計する

これまで作成した各エンティティを接続するための最上位階層エンティティを、VHDL または Verilog HDL で記述します。

1. Quartus® Prime のメニューから **File ▶ New** を選択し、New ダイアログボックスの Design Files ブランチ内から、記述する HDL 用のファイルを選択し、**OK** ボタンをクリックします。

新規ワーキングシートが開きます。

2. Quartus® Prime のメニューから **File ▶ Save As ...** を選択します。

これから作成するデザインファイルを保存するフォルダーを指定し、デザインファイルに名前を付け、**保存(S)** ボタンをクリックします。

【表 2-1-10】 デザインファイルを保存する際の指定内容

項目	指定内容
保存するフォルダー	C:/AlteraFPGA_lab/my_first_fpga
ファイル名	my_first_fpga
ファイルの種類	VHDL で記述 ⇒ VHDL Files (*.vhd *.vhdI)
	Verilog HDL で記述 ⇒ Verilog HDL Files (*.v *.vlg *.verilog)
Add file to current project オプション	On

3. 先ほど保存したワーキングシート (my\_first\_fpga.vhd または my\_first\_fpga.v) に HDL コードを入力します。  
以下に各 HDL の記述サンプルを掲載していますので、参考にしてください。

## 【サンプル 2-1-5】 my\_first\_fpga.vhd コード例

```
library ieee;
use ieee.std_logic_1164.all;

entity my_first_fpga is
  port
  (
    OSC_CLK : in std_logic;
    RESET   : in std_logic;
    BUTTON  : in std_logic;
    LED     : out std_logic_vector (3 downto 0)
  );
end entity;

architecture rtl of my_first_fpga is

  signal pll_c0 : std_logic;
  signal cnt    : std_logic_vector (31 downto 0);
  signal tmp    : std_logic_vector (3 downto 0);

  component pll
    port (
      areset : in std_logic := '0';
      inclk0 : in std_logic := '0';
      c0     : out std_logic;
      locked : out std_logic
    );
  end component;

  component simple_counter
    port (
      clock      : in std_logic;
      reset      : in std_logic;
      counter_out : out std_logic_vector (31 downto 0)
    );
  end component;

  component counter_bus_mux
    port (
      dataa : in std_logic_vector (2 downto 0);
      datab : in std_logic_vector (2 downto 0);
      sel   : in std_logic;
      result : out std_logic_vector (2 downto 0)
    );
  end component;
```

```

begin

pll_inst : pll port map (
    areset  => not RESET,
    inclk0  => OSC_CLK,
    c0      => pll_c0,
    locked  => tmp
);

simple_counter_inst : simple_counter port map (
    clock    => pll_c0,
    reset    => RESET,
    counter_out => cnt
);

counter_bus_mux_inst : counter_bus_mux port map (
    dataa    => cnt(23 downto 21),
    datab   => cnt(26 downto 24),
    sel      => BUTTON,
    result   => tmp(2 downto 0)
);

LED(3)      <= not tmp(3);
LED(2 downto 0) <= not tmp(2 downto 0);

end rtl;

```

### 【サンプル 2-1-6】 my\_first\_fpga.v コード例

```

module my_first_fpga (
    input        OSC_CLK,
    input        RESET,
    input        BUTTON,
    output [3:0] LED
);

wire    pll_c0;
wire [31:0] cnt;
wire [2:0] tmp;

assign LED[2:0] = ~tmp;

pll pll_inst (
    .areset (~RESET),
    .inclk0 (OSC_CLK),
    .c0     (pll_c0),
    .locked (~LED[0])
);

simple_counter simple_counter_inst (
    .clock    (pll_c0),
    .reset    (RESET),
    .counter_out (cnt)
);

```

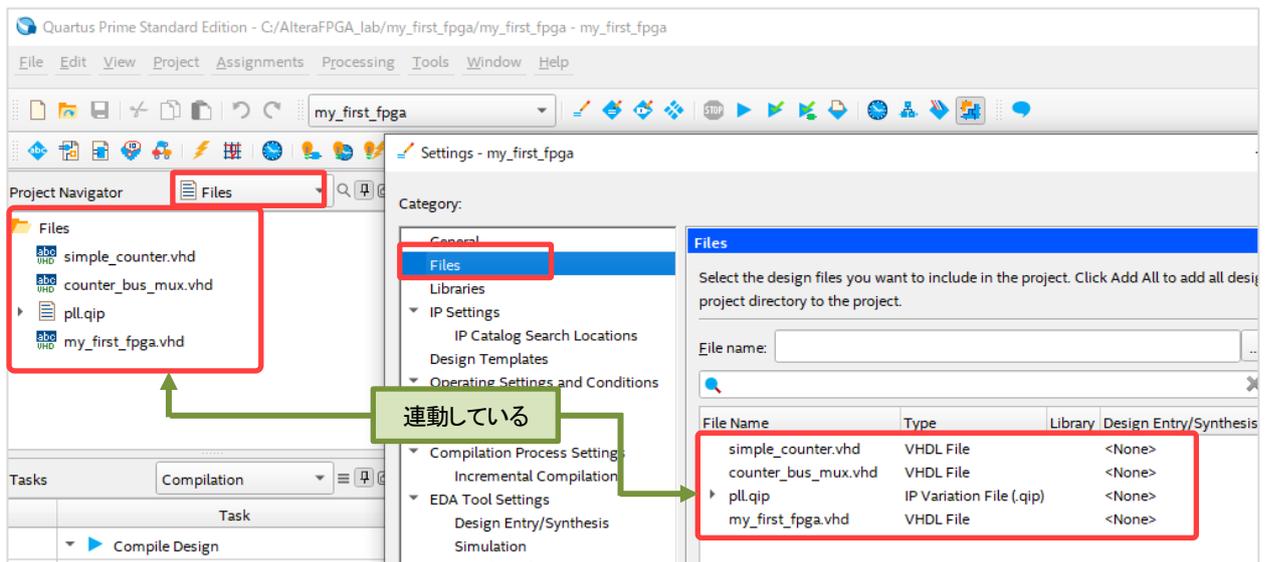
```
counter_bus_mux counter_bus_mux_inst (  
    .dataa (cnt[23:21]),  
    .datab (cnt[26:24]),  
    .sel (BUTTON),  
    .result (tmp)  
);  
  
endmodule
```

4. Quartus® Prime のメニューから **File ▶ Save** を選択し、デザインファイルを保存します。
5. ワーキングシートのタブ右端にある [X] ボタンをクリックし、ファイルを閉じます。

以上で このチュートリアルに必要なすべてのデザイン作成が終了しました。

Project Navigator ウィンドウのプルダウン・リストを Files に切り替え、先ほど作成した 4 つのデザインファイルが現在のプロジェクトに登録されていることを確認します。

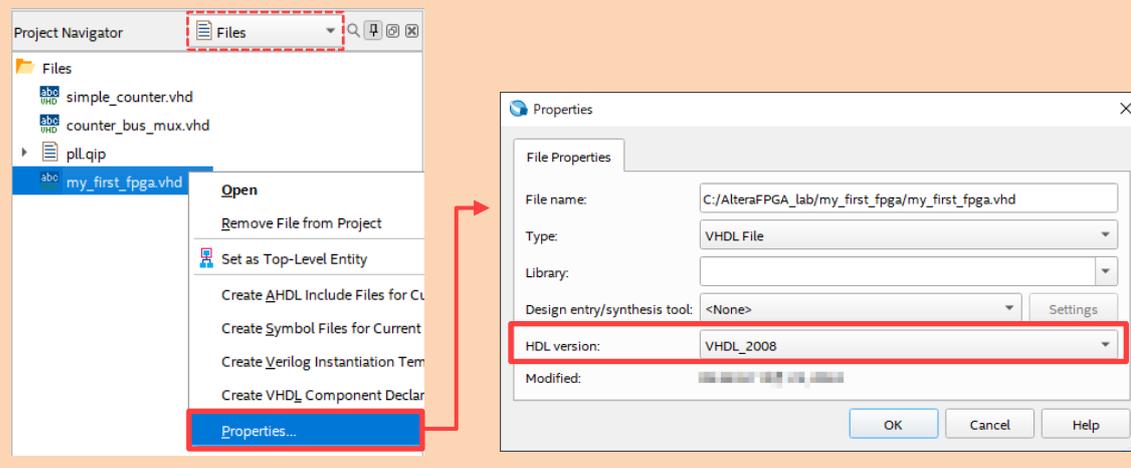
余分なファイルや不足があった場合には、Quartus Prime のメニューから **Project ▶ Add/Remove Files in Project** を選択し、追加あるいは削除を行ってください。



▲ 例: VHDL で設計している場合の Project Navigator / Files (Settings) 画面

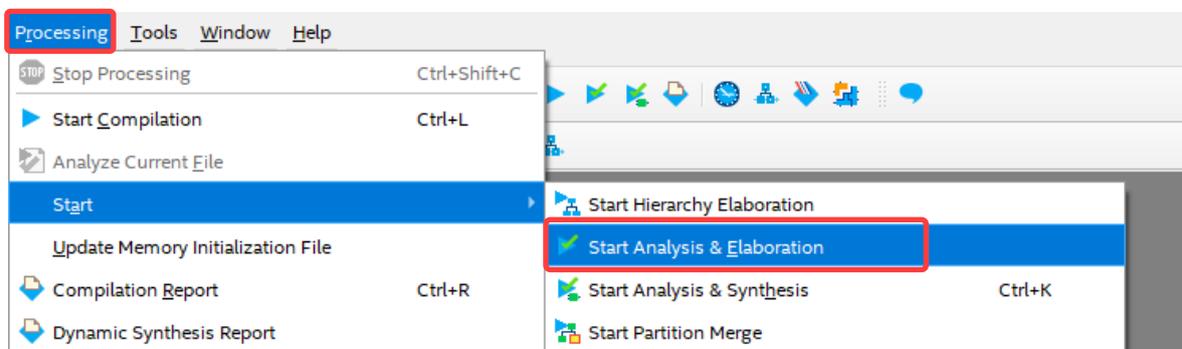
### ⚠ 注記: (VHDL 設計者のみ)

my\_first\_fpga.vhd には VHDL-2008 に対応した記述が含まれています。Project Navigator ウィンドウのビューを Files に切り替え後、ファイルを右クリック選択し、Properties から VHDL\_2008 を選択登録してください。



## 2-3. Analysis and Elaboration を実行する

Quartus® Prime のメニューから **Processing** > **Start** > **Start Analysis & Elaboration** をクリックし、コードのシンタックス・チェックや、デザインのデータベースを作成します。



Messages ウィンドウに、実行が完了したことを示す以下のメッセージが表示されます。

Info: Quartus Prime Analysis & Elaboration was successful.

もしエラーが発生した場合は、HDL コードに問題があります。エラーを解消しないと次のステップには進めません。Messages ウィンドウに表示されたエラー・メッセージの内容をよく読み、指摘されている箇所付近の記述を確認し、記述を修正してください。

また、エラー・メッセージの他、ワーニング・メッセージが発生する場合があります。ワーニングは解消しなくても次のステップへ進めますが、必ず内容を確認し、その内容を回避すべきか無視できるのかをユーザーが判断してください。

各メッセージ内容の詳細を確認するには、ヘルプ機能の活用が有効です (下図参照)。ヘルプには、メッセージが発生した要因や回避するためのヒントが掲載されています。その情報からエラーなどを回避してください。

アイコンをクリックして各メッセージをフィルター。  
右上にメッセージ数を表示。

メッセージを選択して右クリック ▶ Help メニュー

Content List of Messages ↑ Parent topic

**ID:10327 VHDL error at <location>: can't determine definition of operator "<name>" -- found <number> possible definitions**

CAUSE: In a [VHDL Design File \(.vhd\)](#) at the specified location, you used the specified operator. However, Quartus Prime Integrated Synthesis cannot determine the definition of the operator because no definition exists or Quartus Prime Integrated Synthesis found the specified number of different possible definitions for the operator.

ACTION: Make sure the design clearly specifies the definition of the operator.

デザインの修正などコード編集を行った場合は、デザインファイルを保存 (**File** ▶ **Save**) し、再び **Start Analysis & Elaboration** を実行します。エラーやワーニングを回避できたら、ここまでの作業は完了です。

以上で 論理回路の設計は完了です。

### 3. 論理シミュレーションを実行する

2章 で作成した論理回路の動作確認をするため、論理シミュレーションを行います。(この作業は、開発フローの [4] にあたります。)



シミュレーションを行うには、検証するデザインに対して入力する信号と出力した信号をモニターする内容を HDL 言語で表現した “テストベンチ” が必要です。テストベンチは、仕様に合わせてユーザーが設計します。

ここでは、このチュートリアル用にサンプル提供されたテストベンチ・ファイルを使用します。[このチュートリアル \(本資料\) を入手した Web ページ](#) からダウンロードし、プロジェクト・フォルダーに保存してください。

#### 参考:

- テストベンチの設計経験がないユーザーは、以下もご活用ください。
  - ・ [はじめてみよう！テストベンチ](#)

シミュレーション検証するには、Altera® FPGA をサポートしている他社の言語シミュレーターが必要です。

このチュートリアルでは、Questa\* - Intel® FPGA Edition を使用して作業を進めます。(メニューや操作フローは、Questa\* - Intel® FPGA Starter Edition も同様です。)

Questa\* - Intel® FPGA Edition を起動しメニューを選択してオペレーションするフローもありますが、このチュートリアルでは Quartus® Prime Standard Edition の “NativeLink 機能” を活用して Questa\* - Intel® FPGA Edition にシミュレーションを実行させるフローで作業を行います。

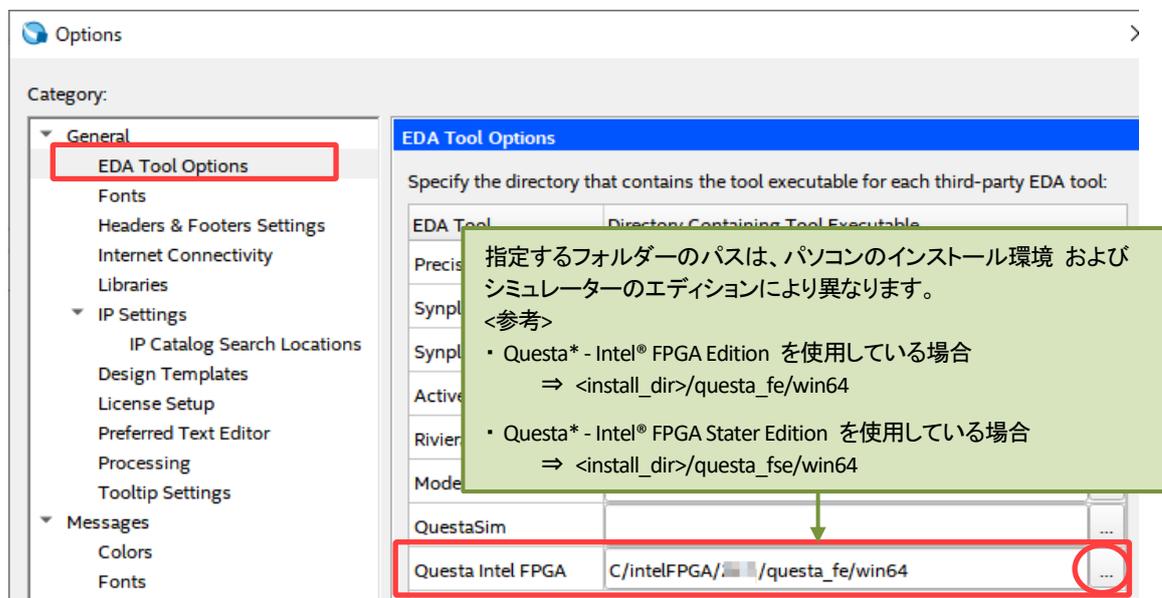
#### 参考:

- NativeLink シミュレーションについては、下記 Web ページを参照してください。
  - ・ [FPGA のファンクション・シミュレーションは NativeLink で解決しよう](#)
- Questa\* - Intel® FPGA Edition を起動して GUI 操作によりシミュレーションを実施する場合は、下記 Web ページのドキュメントを参考にしてください。
  - ・ [ModelSim - Intel FPGA Edition - RTL シミュレーションの方法](#)

#### ① Note:

Questa\* - Intel® FPGA Edition の GUI が起動している場合は、メニューから **File ▶ Quit** を選択し、終了しておきましょう。

1. Quartus® Prime のメニューから **Tools > Options** を選択し、**Category** 内の **General > EDA Tool Options** をクリックします。
2. **Questa Intel FPGA** 行 右横の [...] ボタンをクリックし、Questa\* - Intel® FPGA Edition の実行プログラムが格納されているフォルダーのパスを設定します。

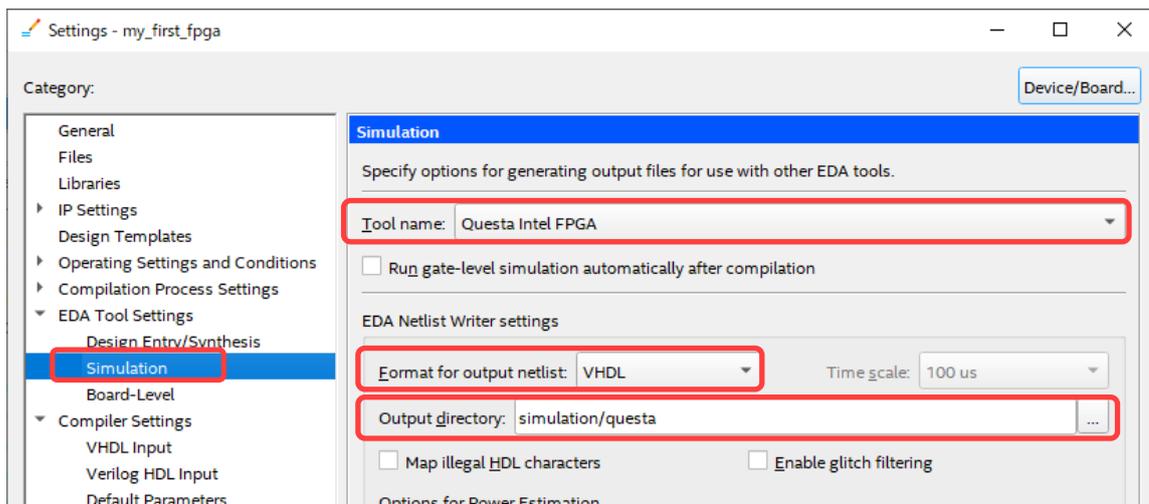


設定後、OK ボタンで Options ダイアログボックスを閉じます。

3. Quartus® Prime のメニューから **Assignments > Settings** を選択し、**Category** 内の **EDA Tool Settings > Simulation** をクリックします。[表 3-1] および次頁の図を参考に、各項目を設定してください。

【表 3-1】 Simulation の設定

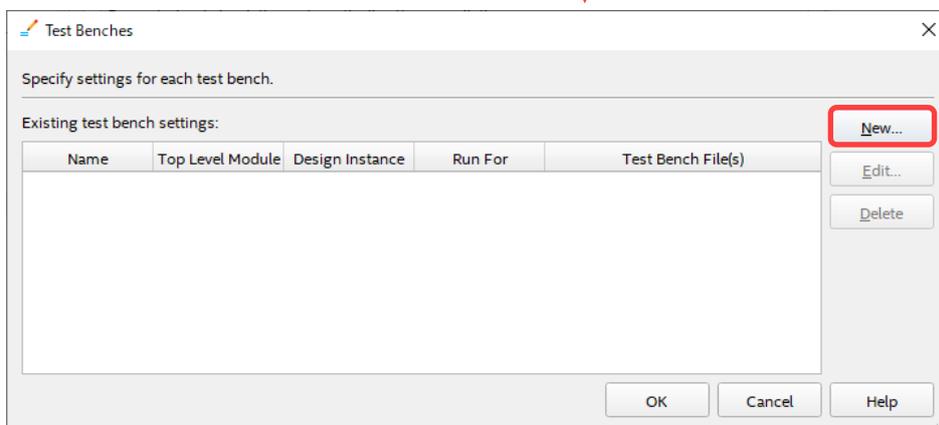
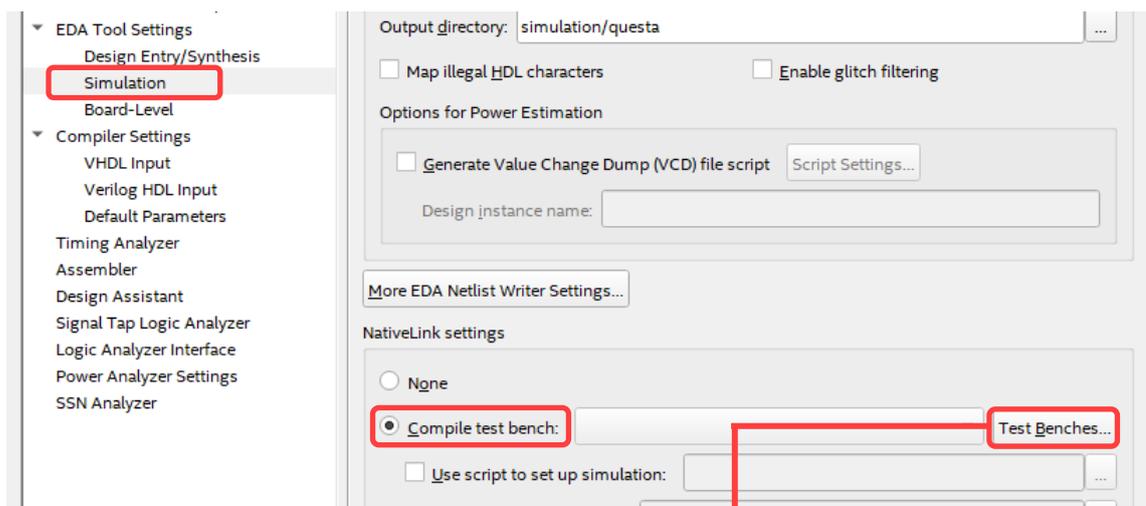
項目	設定内容
<b>Tools name</b>	Questa Intel FPGA
<b>Format for output netlist</b>	VHDL で設計 ⇒ VHDL を選択
	Verilog HDL で設計 ⇒ Verilog HDL を選択
<b>Output directory</b>	simulation/questa
<b>その他の設定</b>	デフォルト



▲ 例: VHDL で設計している場合の Simulation (EDA Tool Settings カテゴリ) 画面

同じウィンドウ内の NativeLink settings エリアにおいて、**Compile test bench** を選択し、**Test Benches** ボタンをクリックします。

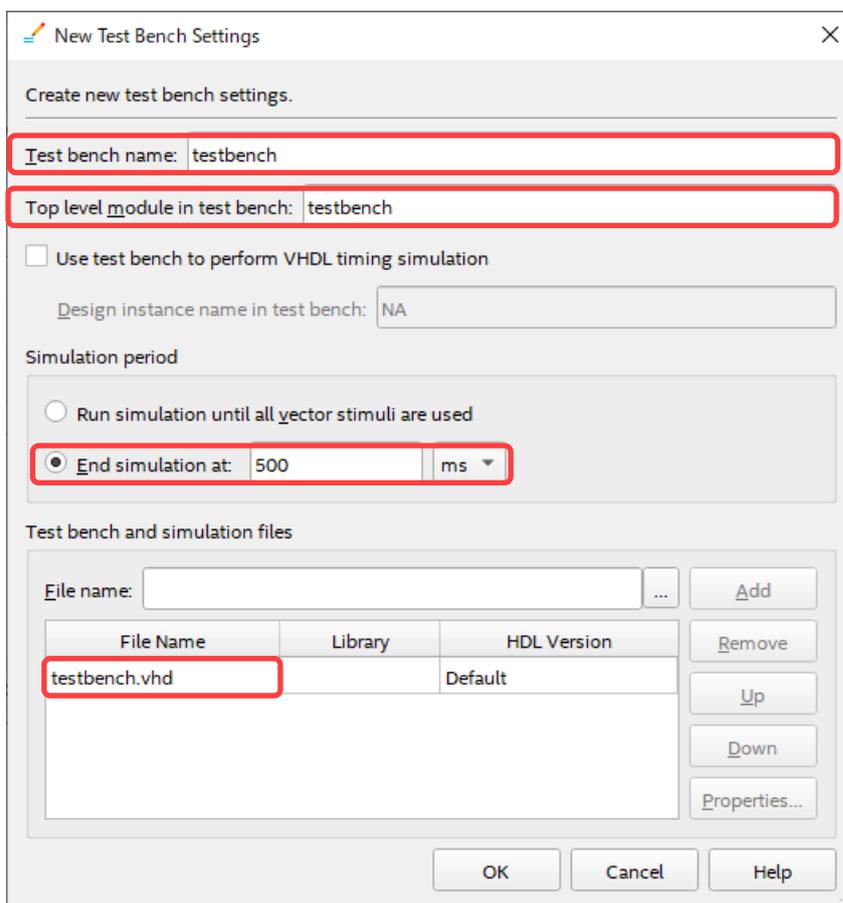
続いて Test Benches ウィンドウの **New** ボタンをクリックします。



New Test Bench Settings ウィンドウにおいて、[表 3-2] および 図を参考に 事前にダウンロードしたテストベンチ・ファイルの情報を入力します。

【表 3-2】 New Test Bench Settings の設定

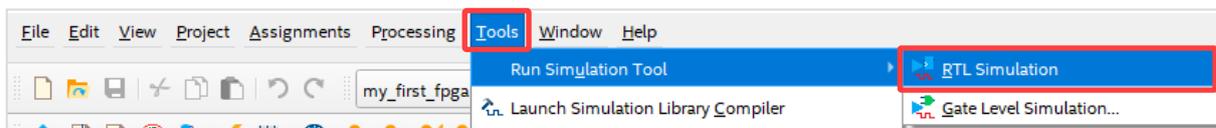
項目	設定内容
<i>Test bench name</i>	testbench
<i>Top level module in test bench</i>	testbench
<i>Simulation period セクション</i>	End simulation at: 500 と入力 / 単位: ms を選択
<i>Test bench and simulation files セクション</i>	ブラウザボタンによりテストベンチ・ファイルを選択、Open ボタンをクリック、Add ボタンをクリックして登録
	VHDL で設計 ⇒ testbench.vhd を選択
	Verilog HDL で設計 ⇒ testbench.v を選択



▲ 例: VHDL で設計している場合の New Test Bench Settings 画面

すべての設定画面を **OK** ボタンで閉じます。

4. Quartus® Prime のメニューから **Tools > Run Simulation Tool > RTL Simulation** を選択し、論理シミュレーション (RTL レベル・シミュレーション) を実行します。



自動的に Questa\* - Intel® FPGA Edition の GUI が起動し、シミュレーションが実行されます。

画面左下に、シミュレーションの実行時間が表示されます。

このチュートリアルでは 500ms の実行が終了するまで待ちます。

```

# Time: 35 ns Iteration: 4 Instance: /testbench/ul/pll_inst/altpll
# ** Note: Cyclone 10 LP PLL locked to incoming clock
# Time: 170 ns Iteration: 3 Instance: /testbench/ul/pll_inst/altpll

```

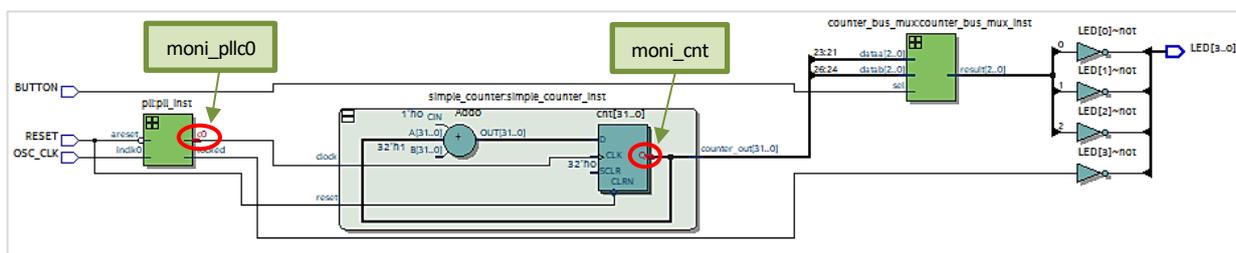
Now: 3,917,235,250 ps Delta: 0 sim:/testbench

もし HDL デザインのシンタックス・エラーなどで処理が停止してしまった場合は、一度 Questa\* - Intel® FPGA Edition の GUI を閉じ、エラーの発生した HDL ファイルを修正および保存してください。その後、再度 Quartus® Prime のメニューから **Tools > Run Simulation Tool > RTL Simulation** を選択し、シミュレーションを実行してください。

5. シミュレーションが終了したら、Wave ウィンドウに表示された波形を確認します。

moni\_pll0、moni\_cnt 信号を確認してください。これらは、my\_first\_fpga エンティティの内部信号をモニターしたものです。

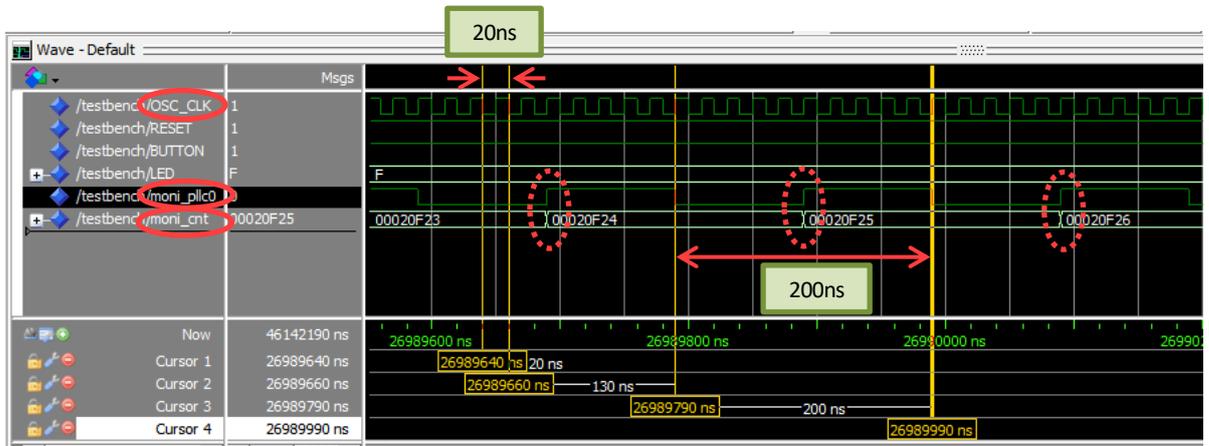
- moni\_pll0 : PLL 回路の出カクロック
- moni\_cnt : カウンター回路の cnt レジスター出力



▲ RTL Viewer (Tools > Netlist Viewers)

このチュートリアルで使用する評価ボードに搭載されているクロックは 50MHz (20ns) です。デザインでは、この 50MHz を FPGA (OSC\_CLK ピン) へ入力し、内部に構築した PLL のリファレンス・クロックに用いています。また、PLL により 5MHz (200ns) のクロックを生成させ、カウンター回路のクロックに使用しています。

Wave ウィンドウにおいて、moni\_pll0 が正常に生成されていることが確認できます。同様に、moni\_cnt が PLL の出力クロックに同期して正常にカウントアップしていることが確認できます。(論理シミュレーションは遅延を考慮しないため、クロック信号のエッジで出力信号が変化していることが確認できます。)

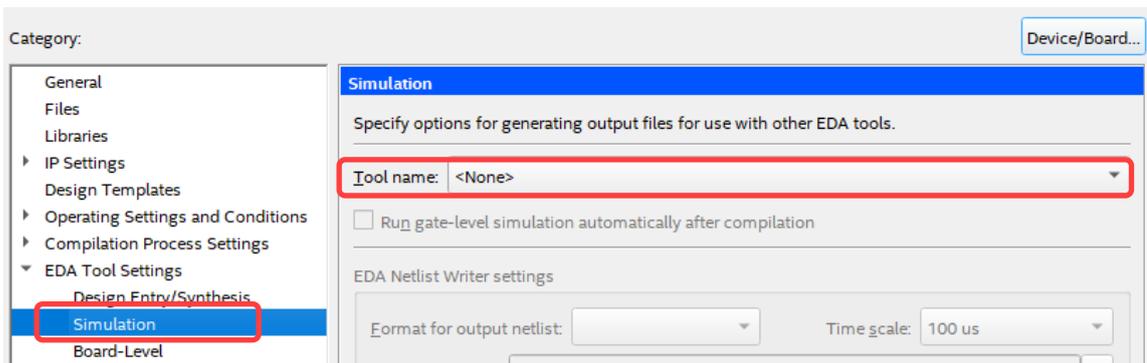


6. Questa\* - Intel® FPGA Edition のメニューから **Simulate > End Simulation** を選択し、シミュレーションを終了します。
7. Questa\* - Intel® FPGA Edition のメニューから **File > Quit** を選択し、GUI を閉じます。

以上で、作成したデザインの論理シミュレーションは終了です。

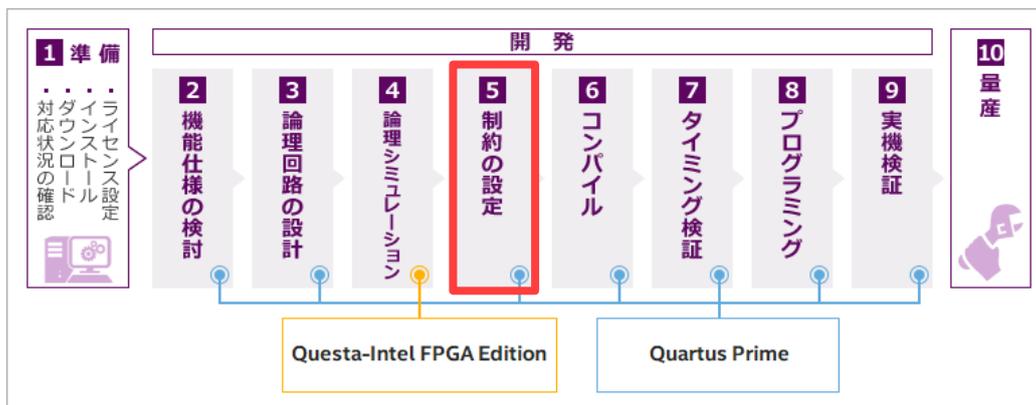
以降にシミュレーション作業を行いませんので、以下の設定を無効にします。

8. Quartus® Prime のメニューから **Assignments > Settings** を選択し、Category 内の **EDA Tool Settings > Simulation** をクリックします。**Tools name** において、リストから **<None>** を選択後 Settings ダイアログボックスを **[OK]** ボタンで閉じます。



## 4. 制約の設定をする

論理回路に対して各種オプションを設定します。(この作業は、開発フローの [5] にあたります。)



このチュートリアルでは、以下の内容に制約を行います。

- 論理回路の入出力ピンにターゲット・デバイスの I/O ピン番号へ割り当て、ボード仕様に応じた I/O 規格を設定 [ デザインのピンアサイン ]
- 論理回路に対して希望の動作スピードを設定 [ タイミング制約 ]

それ以外の制約は、Quartus® Prime のデフォルト制約を適用させます。

### 4-1. デザイン上のピンをデバイスにアサインする

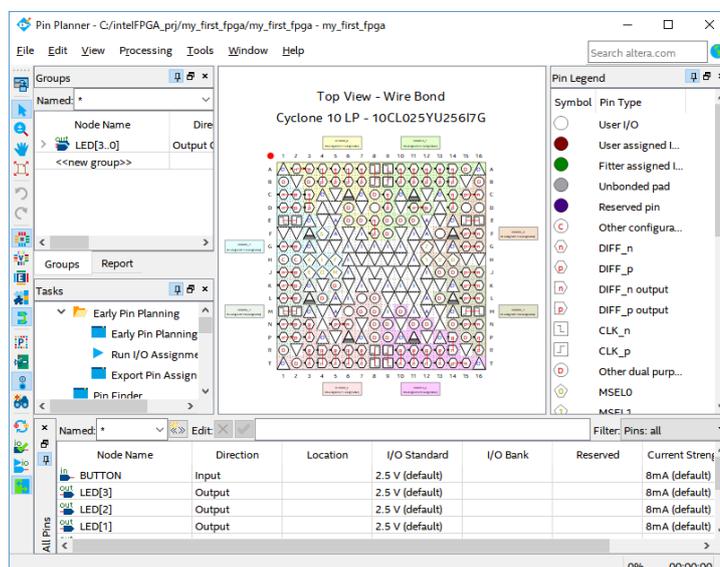
ピンの割り当てを行うには、Analysis & Elaboration の実行が必要です。

(このチュートリアルでは [2-3 章](#) において すでに実行していますので、この作業はスキップできます。未完了の場合には、以下のメニューにより実行してください。)

Quartus® Prime のメニューから **Processing > Start > Start Analysis & Elaboration** を選択し実行します。

Analysis & Elaboration 実行後にエラーが発生した場合は、[2-3 章](#) を参考にエラーを回避してください。

1. Quartus® Prime のメニューから **Assignments > Pin Planner** を選択し、Pin Planner を起動します。



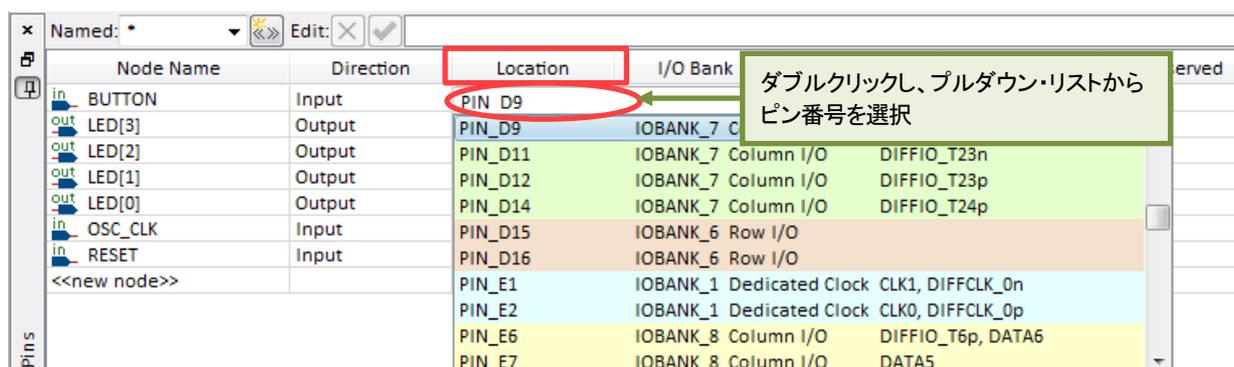
- Pin Planner の All Pins リストに、デザインで使用しているピンが表示されていることを確認してください。各ピン名の行にある Location カラムに、[表 4-1-1] のとおり ピン番号を指定します。

【表 4-1-1】ピン番号アサイン表

ピン名	Cyclone® 10 LP Evaluation Kit	MAX® 10 FPGA Evaluation Kit
BUTTON	D9	120
LED[3]	J13	140
LED[2]	J14	135
LED[1]	K15	134
LED[0]	L14	132
OSC_CLK	E1	27
RESET	E15	121

① Note:

各ピン番号が、使用する評価ボードのどのタクトスイッチや DIP スイッチ、LED なのかを、評価ボードのリファレンス・マニュアルや回路図で確認してください。



▲ 例: Cyclone® 10 LP Evaluation Kit の場合

- Pin Planner の All Pins リストにおいて、各ピン名の行にある I/O Standard カラムに、[表 4-1-2] のとおり I/O 規格を指定します。

【表 4-1-2】I/O 規格アサイン表

ピン名	Cyclone® 10 LP Evaluation Kit	MAX® 10 FPGA Evaluation Kit
BUTTON	3.3-V LVTTTL	3.3-V LVTTTL
LED[3]	3.3-V LVTTTL	3.3-V LVTTTL
LED[2]	3.3-V LVTTTL	3.3-V LVTTTL
LED[1]	3.3-V LVTTTL	3.3-V LVTTTL
LED[0]	3.3-V LVTTTL	3.3-V LVTTTL
OSC_CLK	3.3-V LVTTTL	3.3-V LVTTTL
RESET	3.3-V LVTTTL	3.3-V LVTTTL

### 目 参考:

- 以下に、ここでの作業に関連したドキュメントをご案内します。ご参考ください。
  - ・ [「Quartus Prime はじめてガイド - ピン・アサインの方法」](#)
  - ・ [「Quartus Prime はじめてガイド - デバイス・オプションの設定方法」](#)

以上で デザインのピンアサインは終了です。

## 4.2. タイミング制約

論理回路に対して希望の動作スピードを設定するため、タイミング制約を作成します。

Quartus® Prime では、ASIC 業界で標準的に使用されている Synopsys Design Constraints (SDC) ファイル・フォーマットでタイミング制約を行います。

1. Quartus® Prime のメニューから **Tools** ▶ **Timing Analyzer** を選択し、Timing Analyzer ウィンドウを表示します。
2. Timing Analyzer ウィンドウのメニューから **File** ▶ **New SDC File** を選択すると、Quartus® Prime のメイン・ウィンドウに SDC ファイル・エディタが開きます。
3. SDC ファイルに、下記コードを入力してください。

【サンプル 4-2-1】SDC コード

```
create_clock -period 20.000 -name osc_clk OSC_CLK
derive_pll_clocks
derive_clock_uncertainty
```

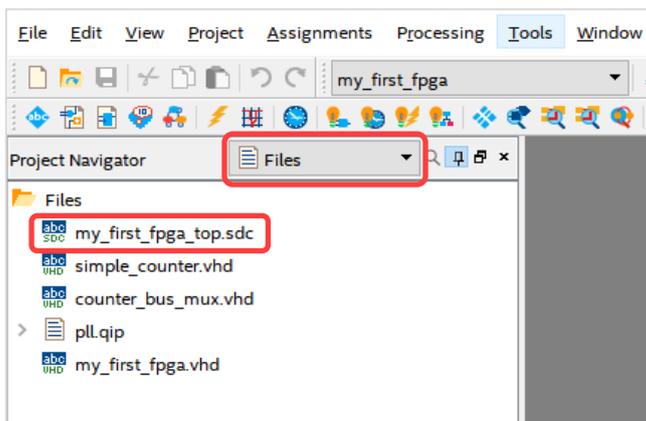
4. Quartus® Prime のメニューから **File** ▶ **Save As...** を選択し、SDC ファイルに名前を付けて保存します。

【表 4-2-1】デザインファイルを保存する際の指定内容

項目	指定内容
保存するフォルダー	C:/AlteraFPGA_lab/my_first_fpga
ファイル名	my_first_fpga_top.sdc
ファイルの種類	Synopsys Design Constraints Files (*.sdc)
Add file to current project オプション	On

5. sdc ファイルのタブ右端にある [X] ボタンをクリックし、ファイルを閉じます。

6. Timing Analyzer ウィンドウのメニューから **File > Close** を選択し、ウィンドウを閉じます。
7. Project Navigator ウィンドウのプルダウン・リストを Files に切り替え、my\_first\_fpga\_top.sdc が現在のプロジェクトに登録されていることを確認します。



#### 参考:

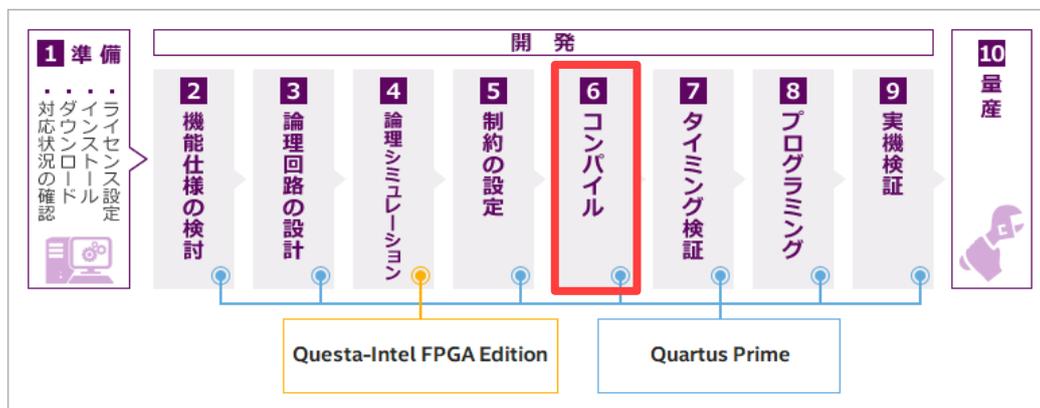
- 以下に、ここでの作業に関連したドキュメントをご案内します。ご参考ください。
  - ・ [「Quartus Prime はじめてガイド - TimeQuest によるタイミング制約の方法」](#)

以上で タイミング制約は終了です。

これで、デザインに対する制約設定は完了しました。

## 5. コンパイルを実行する

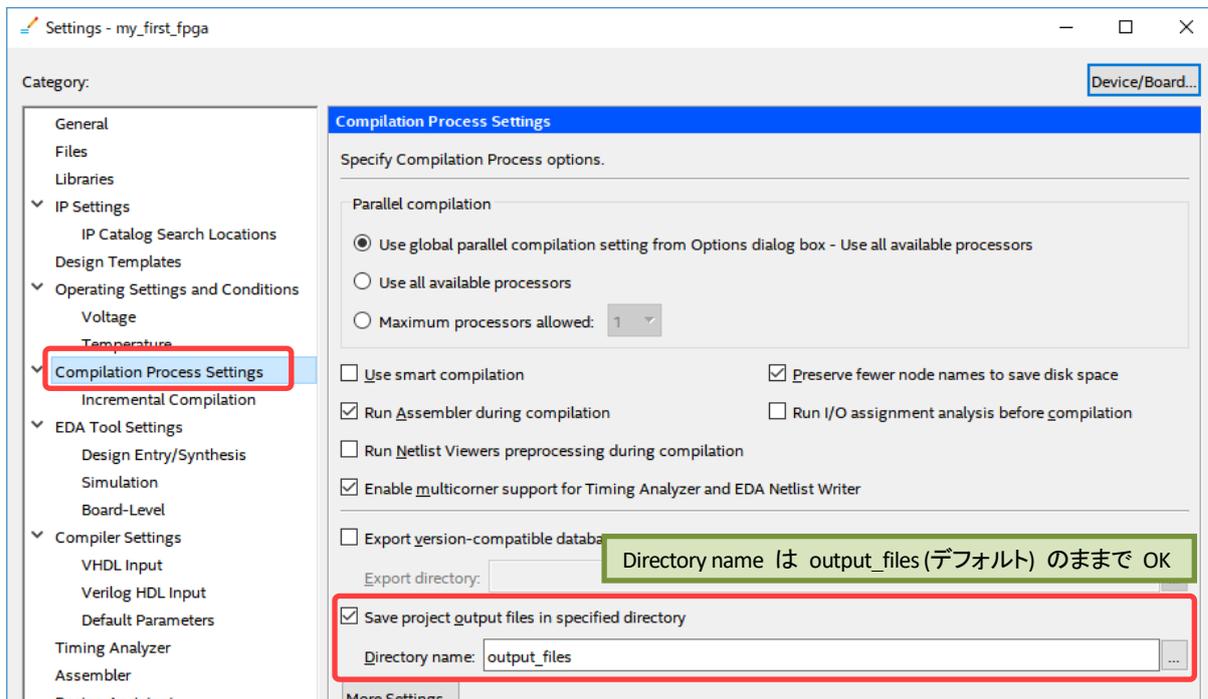
完成したデザインをコンパイルします。(この作業は、開発フローの [6] にあたります。)



コンパイルにより生成されるファイル (コンパイル・レポートやプログラミング・ファイルなど) を保存するフォルダーを別途プロジェクト・フォルダー内に作成します。

この作業は必須ではありませんが、プロジェクト・フォルダーが整理され管理しやすくなります。

Quartus® Prime のメニューから **Assignments > Settings > Compilation Process Settings** を選択し、**Save project output files in specified directory** オプションを有効にします。



1. Quartus® Prime のメニューから **Processing > Start Compilation** を選択し、コンパイルを実行します。

コンパイルは、論理合成や配置配線、タイミング解析、プログラミング・ファイル作成が自動で行われます。

コンパイルを実行すると、Messages ウィンドウに様々な情報が表示されます。これらメッセージに関しては、[2-3 章](#) を参考に対処してください。

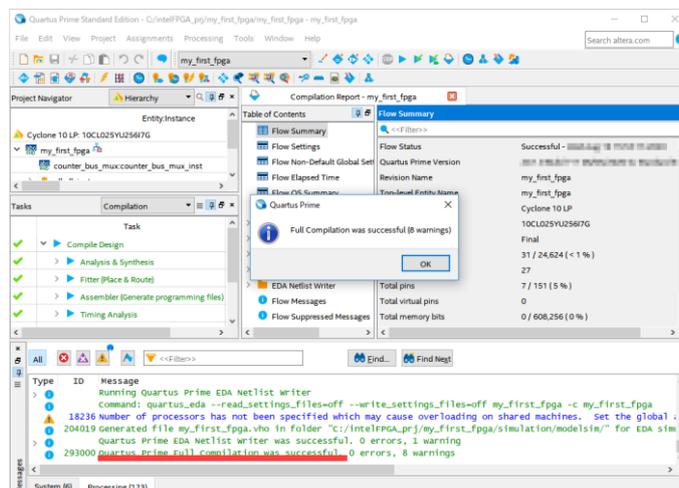
ツールバーのアイコンで実行するときはこのボタン！



2. Messages ウィンドウに *Quartus Prime Full Compilation was successful* のメッセージが確認できたら、コンパイル成功です。

コンパイルが終了すると、コンパイル・レポート (Compilation Report) が自動ポップアップし、コンパイル結果が表示されます。実施された Analysis & Synthesis, Fitter, Assembler, Timing Analyzer の工程ごとに、レポートが生成されます。ユーザーは、自分の仕様に適切な処理が行われたかを確認するため、コンパイル・レポートを必ず見る必要があります。

詳細は、次のドキュメントをご覧ください。



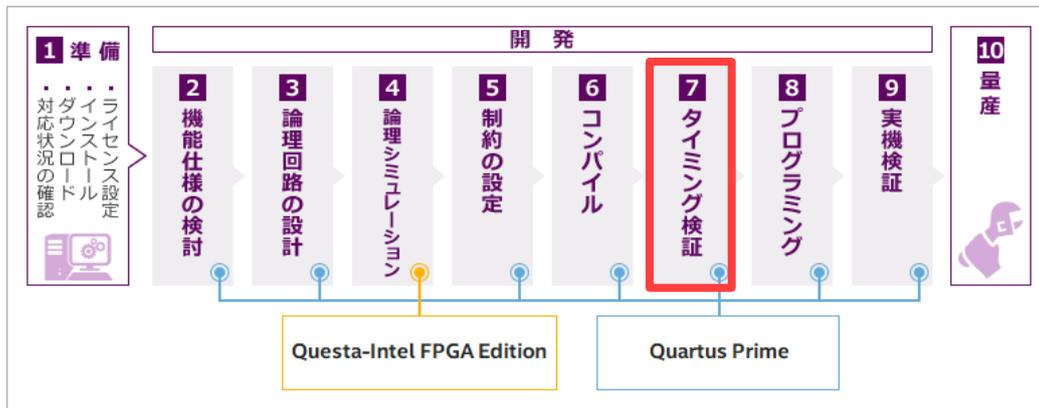
目 参考:

- 以下に、ここでの作業に関連したドキュメントをご案内します。ご参考ください。
  - ・ [「Quartus Prime はじめてガイド - コンパイル・レポート・ファイルの見方」](#)

コンパイル・レポートを閉じた後に再び表示させるには、Quartus® Prime のメニューから **Processing** ➤ **Compilation Report** を選択してください。

## 6. タイミングを検証する

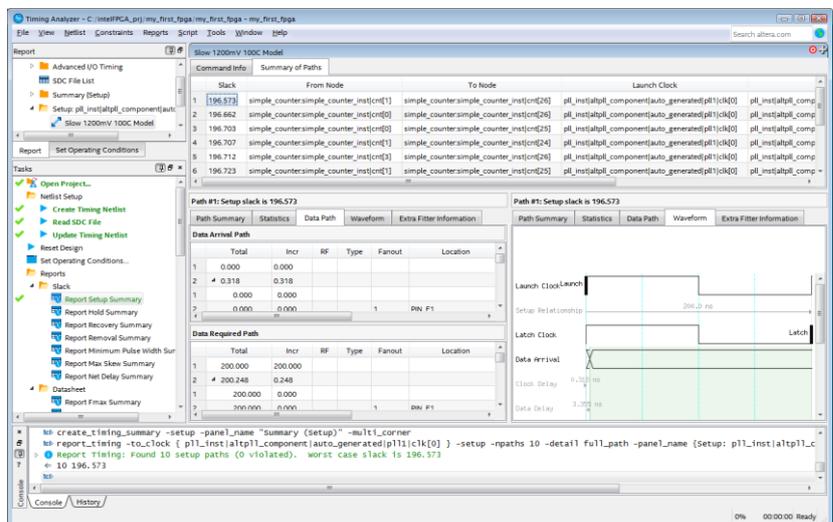
コンパイルの配置配線結果が、ユーザーのタイミング要件を満足したかを確認します。(この作業は、開発フローの [7] にあたります。)



Quartus® Prime は、コンパイル実行と同時に、タイミング検証も実施しています。ユーザーが事前に登録したタイミング制約と配置配線した際のタイミング情報とを比較し、ユーザーの要求値を満足 (メット) しているかどうかを Timing Analyzer レポートで確認することができます。Timing Analyzer レポートの結果がユーザーの要求値を満足していない場合、実機でも動作を保証しないため、タイミング検証は非常に重要な作業工程です。

まずコンパイル・レポート内の Timing Analyzer レポートで検証結果のサマリーを確認し、タイミングがメットできていない箇所があった場合は、そのパスを特定します。その後 Timing Analyzer を別途起動して要因を追究し、回避しなければなりません。

そのため、クロックに対する同期/非同期の考え方や、レジスタに対するセットアップタイムやホールドタイムと言った、静的タイミング解析に重要な知識を要します。



このチュートリアルは FPGA 開発のビギナーを対象としており、プロジェクト・デザインはタイミング制約を満足する結果が得られる仕様のため、タイミング検証は割愛します。Quartus® Prime の操作に慣れてきたら、次のステップとして習得してください。

#### 参考:

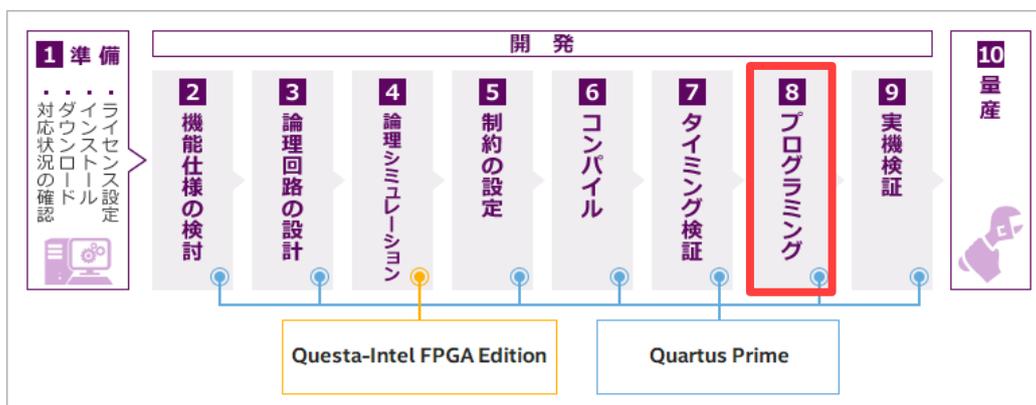
- 以下に、ここでの作業に関連したドキュメントをご案内します。ご参考ください。
  - ・ [「Quartus Prime はじめてガイド - TimeQuest によるタイミング解析の方法」](#)

#### ① Note:

- メーカー公式 FPGA テクニカル・トレーニング (有償)
  - ・ [「Quartus Prime パーフェクト・コース II : タイミング解析」](#)
- 弊社オリジナル [FPGA のタイミング関連 オンラインセミナー 各種](#)

## 7. プログラミングをする

評価ボード上の FPGA を実際に動作させる準備が整ったので、作成したファイルを FPGA へダウンロードします。(この作業は、開発フローの [8] にあたります。)



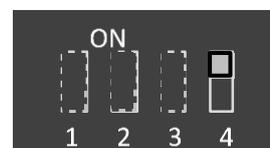
#### ⚠ 注記:

実際の開発では、コンパイル後に Quartus® Prime の Timing Analyzer によるタイミング検証を行い、期待どおりの動作が実現できるかを検証します。期待するタイミングを満足できることが確認できたら、ボード上のデバイスヘータを書き込みます。

今回の演習ではタイミング検証を省略していますが、自身の開発時は必ずタイミング検証を行った上でデバイスへの書き込みを行ってください。

#### ① Note:

Cyclone® LP Evaluation Kit をご利用の場合は、事前に DIP スイッチ [SW1] のうち SW1.4 を右図のとおりに変更してから作業を行ってください。



プログラミング・データは、コンパイル実行時に作成されます。

FPGA へ書き込む場合は sof ファイルです。sof ファイルは、FPGA の SRAM 部分に書き込むための形式です。SRAM にデータを書き込んだ場合、FPGA へ供給される電源が OFF になるとデータは消去されます。そのため、開発ボードの電源を OFF にしても次回の電源投入時に FPGA が動作をするためには、インテル FPGA 用のコンフィグレーション ROM (FPGA が起動するためのブート ROM) に開発したデザインのデータを書き込む必要があります。コンフィグレーション ROM へ書き込む場合は pof ファイルです。

コンフィグレーション ROM へ書き込んでから実機検証も行えますが、FPGA へのデータ書き込み時間と比べてコンフィグレーション ROM への書き込みには時間がかかることと、書き込み/書き換え/消去に対する実行回数に上限があるため、一般的には先に FPGA の実機動作検証を行い、修正が無いことを確認した上でコンフィグレーション ROM にプログラムすることを推奨しています。

このチュートリアルにおいても、まずは FPGA に sof ファイルを書き込み、実機検証を行った後にコンフィグレーション ROM へ書き込みを行います。

## 7-1. sof ファイルを書き込む

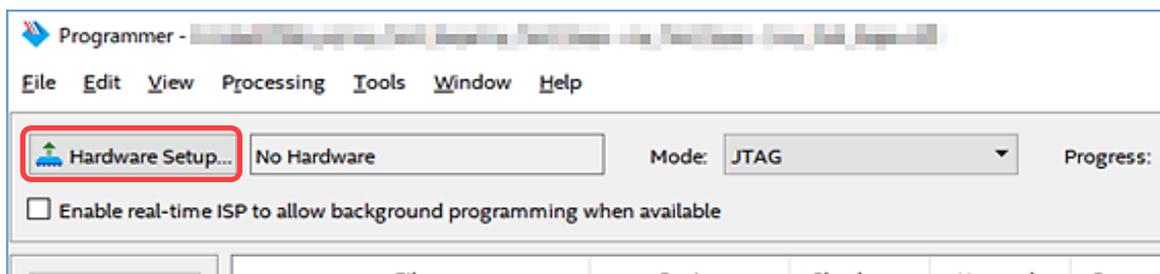
1. ダウンロード・ケーブル、パソコン、評価ボードを接続します。

ボード上の JTAG コネクタの位置や向きは、各評価ボードのユーザーガイドや回路図をご確認ください。

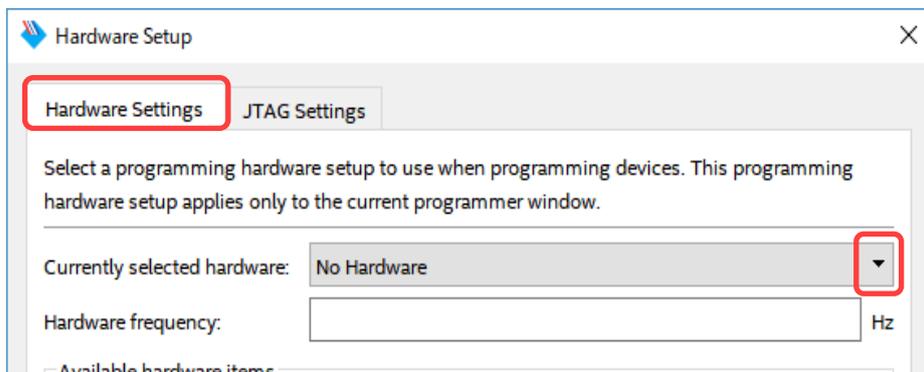
【表 7-1-1】ダウンロード・ケーブルの接続

評価ボード	説明
Cyclone® 10 LP Evaluation Kit	<ol style="list-style-type: none"> <li>1. Cyclone® 10 LP Evaluation Kit に付属しているケーブルにおいて、mini USB (Type-B) を評価ボードに接続します。</li> <li>2. USB ケーブルのうち黒いコネクタをパソコンへ接続してください。(ダウンロード・ケーブルと電源供給ケーブルを兼用しています。)</li> <li>3. ボードに電源が供給されます。</li> </ol>
MAX® 10 FPGA Evaluation Kit	<ol style="list-style-type: none"> <li>1. インテル FPGA ダウンロード・ケーブル II 本体と USB ケーブルを接続します。</li> <li>2. インテル FPGA ダウンロード・ケーブル II 本体から出ているケーブルの 10 pin ヘッダーを評価ボード上の JTAG コネクタに接続します。</li> <li>3. 電源用 USB ケーブルで、パソコンと評価ボードを接続します。(ボードに電源が供給されます。)</li> </ol>

2. Quartus® Prime のメニューから **Tools > Programmer** を選択し、Programmer ウィンドウを起動します。
3. Programmer ウィンドウの **Hardware Setup** ボタンをクリックします。



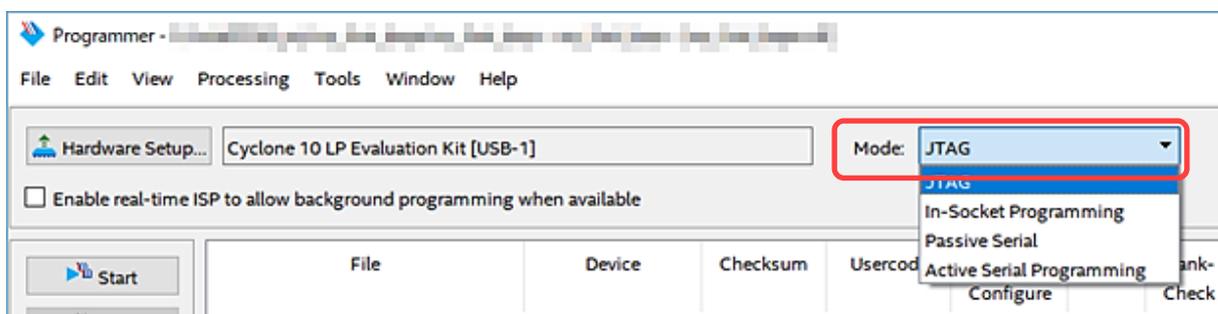
4. **Hardware Settings** タブの **Currently selected hardware** 項のプルダウン・リストから、使用するダウンロード・ケーブルを選択し、**Close** ボタンをクリックします。



【表 7-1-2】 Currently selected hardware に選択するケーブル名

評価ボード	Currently selected hardware に検出される名前
Cyclone® 10 LP Evaluation Kit	Cyclone 10 LP Evaluation Kit
MAX® 10 FPGA Evaluation Kit	使用するダウンロード・ケーブル名 (例: USB-Blaster II)

5. Programmer ウィンドウの **Mode** プルダウン・リストより、**JTAG** を選択します。



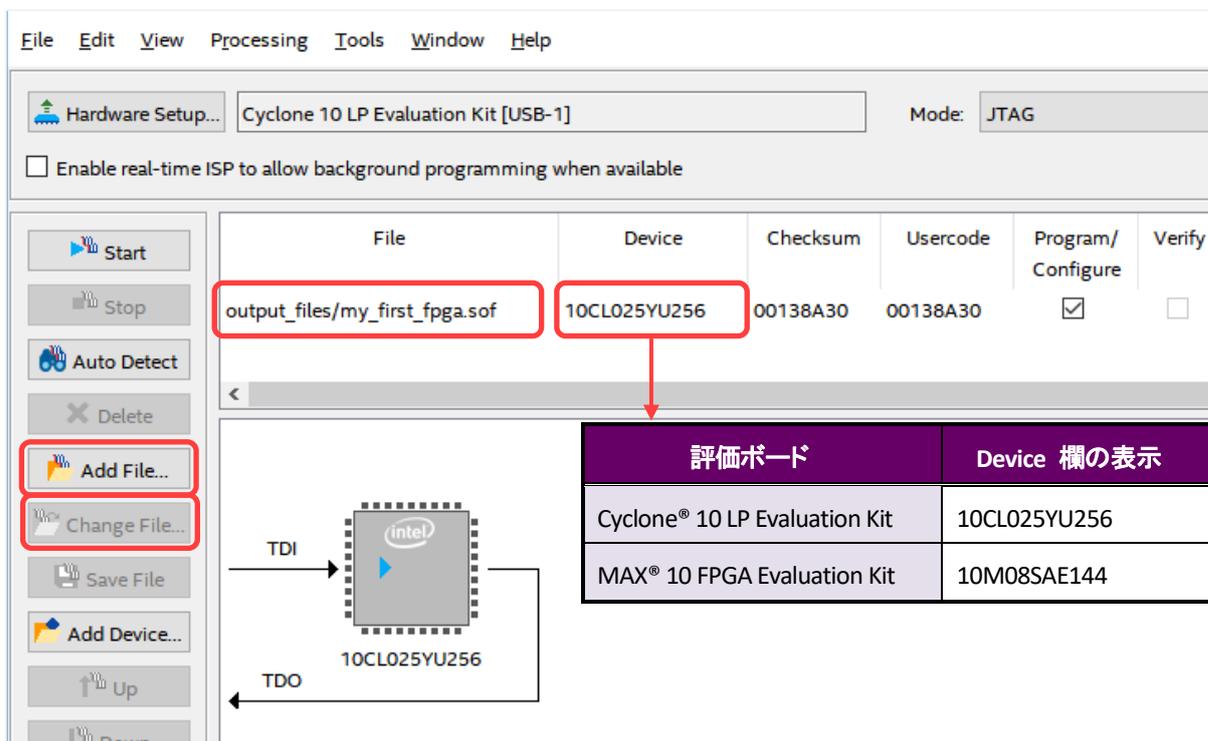
▲ 例: Cyclone® 10 LP Evaluation Kit の場合

① Note:

- ・ Cyclone® 10 LP Evaluation Kit は、オンボード インテル FPGA ダウンロード・ケーブル II 対応基板です。よって、ご利用前に [インテル FPGA ダウンロード・ケーブル II のドライバをインストール](#) してください。
- ・ インテル FPGA ダウンロード・ケーブル II は、TCK 周波数を変更可能です。詳細は、下記 Web ページをご参考ください。

[USB-Blaster™ II の TCK 周波数を変更してみよう](#)

6. Programmer ウィンドウの File 欄にはプロジェクト用の *my\_first\_fpga.sof* ファイルがすでに登録され、Device 欄にはプロジェクトで指定した FPGA の型番が表示されています。

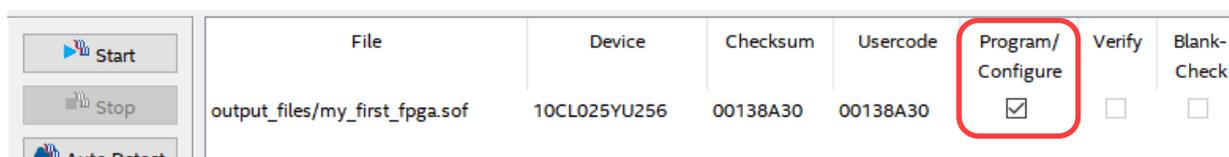


▲ 例: Cyclone® 10 LP Evaluation Kit の場合

もし sof ファイルが未登録の場合は、**Add File** ボタンをクリックし、~~¥~~作業フォルダー¥*output\_files*¥*my\_first\_fpga.sof* を選択後、**Open** ボタンをクリックして File 欄に登録します。

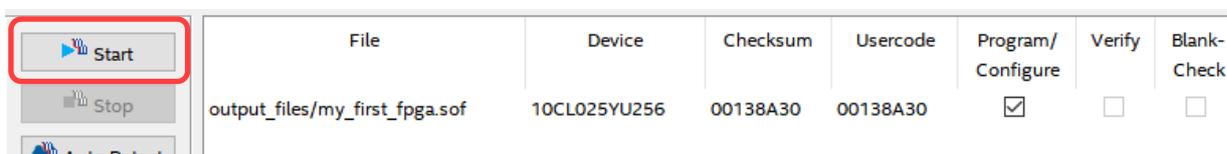
あるいは、別の sof ファイルが登録されている場合には、そのファイルを選択後に **Change File** ボタンをクリックして ~~¥~~作業フォルダー¥*output\_files*¥*my\_first\_fpga.sof* を指定し直します。

7. Programmer ウィンドウの *my\_first\_fpga.sof* 行の右側にある **Program/Configure** オプションにチェックを入れます。



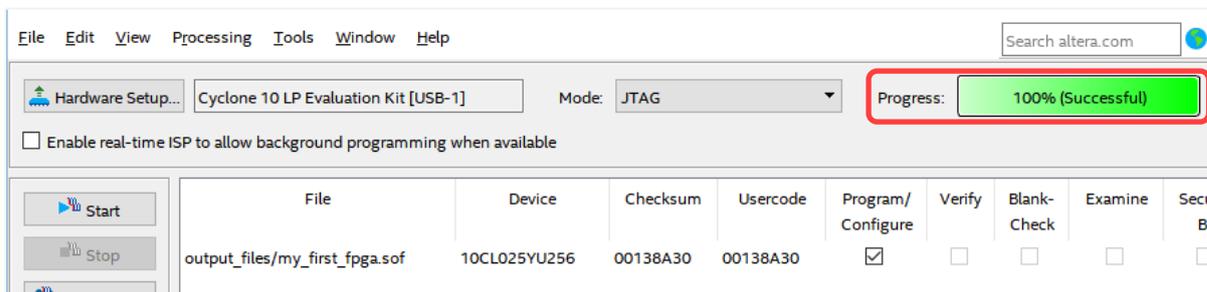
▲ 例: Cyclone® 10 LP Evaluation Kit の場合

8. Programmer ウィンドウの **Start** ボタンをクリックし、FPGA へのダウンロードを開始します

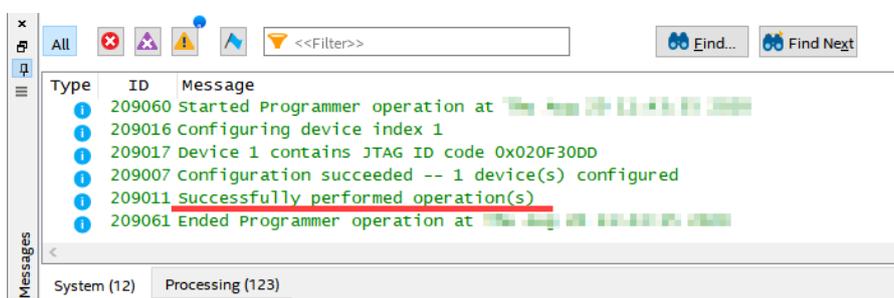


▲ 例: Cyclone® 10 LP Evaluation Kit の場合

Programmer ウィンドウの Progress バーが 100% になり、Quartus® Prime の Messages ウィンドウ (System タブ) に **Info: Successfully performed operation(s)** のメッセージを確認できたら、データのダウンロードは完了です。



▲ 例: Cyclone® 10 LP Evaluation Kit の場合



▲ Messages ウィンドウ

[8章 実機で動作検証をする](#) を参考に、FPGA の動作を確認します。

## 7-2. コンフィグレーション ROM に書き込む

### ⚠ 注記:

この作業は、先に sof ファイルによる実機動作検証 ([7-1 章](#)) を行い、FPGA の動作に修正の必要が無いことを確認した上で行ってください。

### 7-2-1. コンフィグレーション ROM 用のプログラミング・ファイルを作成する

使用する開発ボードに応じたコンフィグレーション ROM 用のプログラミング・ファイルを sof ファイルから作成します。

【表 7-2-1-1】 評価ボードに対応したコンフィグレーション ROM とファイル

評価ボード	コンフィグレーション ROM	作成するファイル
Cyclone® 10 LP Evaluation Kit	EPCQ64	.jic
	EPCQ128A	
MAX® 10 FPGA Evaluation Kit	MAX 10 内蔵 CFM	.pof

## ① Note:

- ・ Cyclone® 10 LP Evaluation Kit に搭載されているコンフィグレーション ROM は、ボード・リビジョンにより異なります。使用する基板がどの ROM に対応しているか、[Cyclone 10 LP Evaluation Kit ユーザーガイド](#)で確認してください。
- ・ EPCQ64 / EPCQ128A は、基板の配線接続仕様に依じて pof ファイルにも対応しています。(Cyclone® 10 LP Evaluation Kit の接続仕様は、pof ファイル・プログラミングに非対応)
- ・ MAX® 10 は、内蔵の CFM 領域にコンフィグレーション・データをプログラムします。

1. Quartus® Prime のメニューから **File** ➤ **Convert Programming Files** を選択し、起動します。
2. Convert Programming Files ダイアログボックスの **Output programming file** 枠内を設定します。

【表 7-2-1-2】 Output programming file

項目	設定内容	
	Cyclone® 10 LP Evaluation Kit	MAX® 10 FPGA Evaluation Kit
<b>Programming file type</b>	JTAG Indirect Configuration File (.jic)	Programmer Object File (.pof)
<b>Mode</b>	Active Serial	Internal Configuration
<b>Configuration device</b>	EPCQ64      EPCQ128A	—
<b>File Name</b>	output_files/my_first_fpga.jic	output_files/my_first_fpga.pof

Output programming file

Programming file type: JTAG Indirect Configuration File (.jic)

Options/Boot info... Configuration device: EPCQ128A ... Mode: Active Serial

File name: output\_files/my\_first\_fpga.jic

Advanced... Remote/Local update difference file: NONE

Create Memory Map File (Generate my\_first\_fpga.map)

Create CvP files (Generate my\_first\_fpga.periph.jic and my\_first\_fpga.core.rbf)

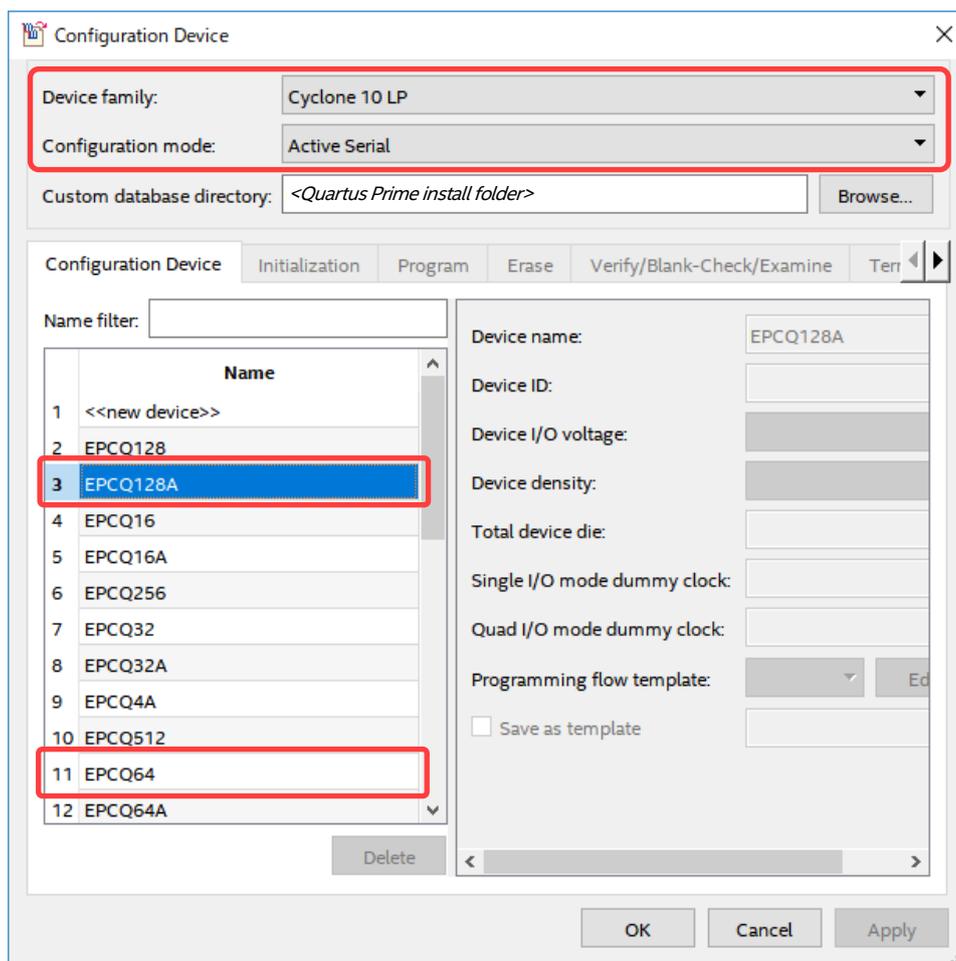
Create config data RPD (Generate my\_first\_fpga\_auto.rpd)

▲ 例: Cyclone® 10 LP Evaluation Kit / EPCQ128A の場合

Cyclone® 10 LP Evaluation Kit ユーザーが Convert Programming Files においてコンフィグレーション ROM を選択する際は、右横のブラウザボタンをクリックし、希望の型番を選択します。([表 7-2-1-3] 参照)

【表 7-2-1-3】 Configuration Device ダイアログボックス

項目	設定内容
Device family	Cyclone 10 LP
Configuration mode	Active Serial
Configuration device	EPCQ64 または EPCQ128A

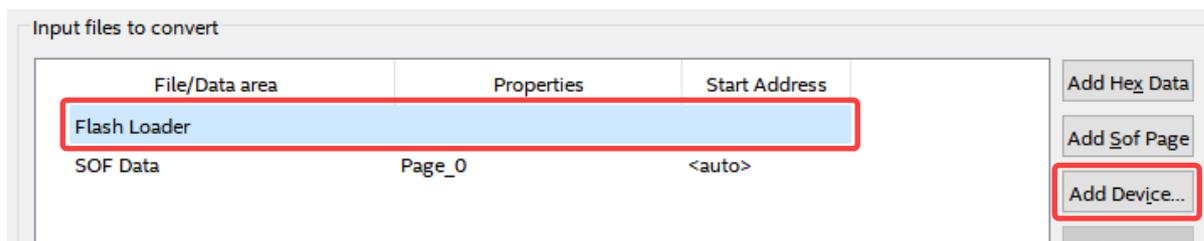


▲ 例: コンフィグレーション ROM を選択 (Cyclone® 10 LP Evaluation Kit)

3. Convert Programming Files ダイアログボックスの **Input files to convert** 枠内を設定します。

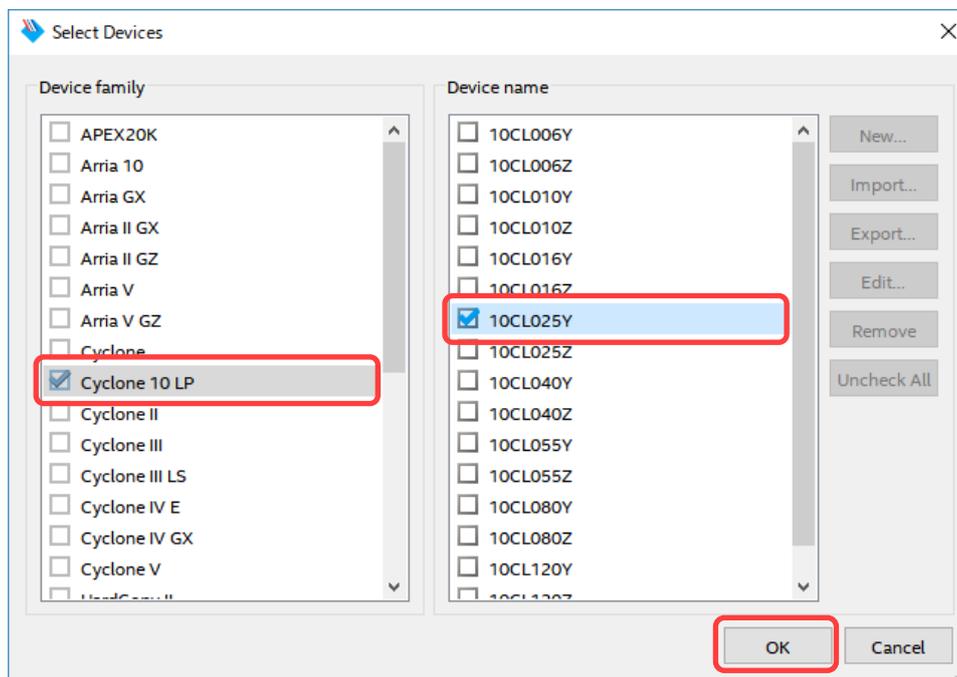
① (Cyclone® 10 LP Evaluation Kit ユーザーのみ作業してください。)

**Flash Loader** 行をハイライト選択し、**Add Device** ボタンをクリックします。

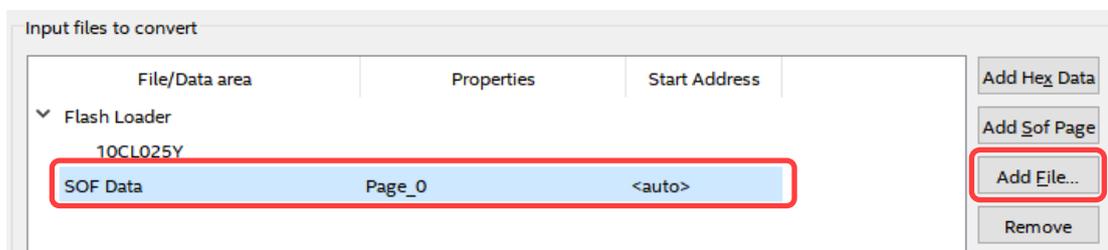


## ② (Cyclone® 10 LP Evaluation Kit ユーザーのみ作業してください。)

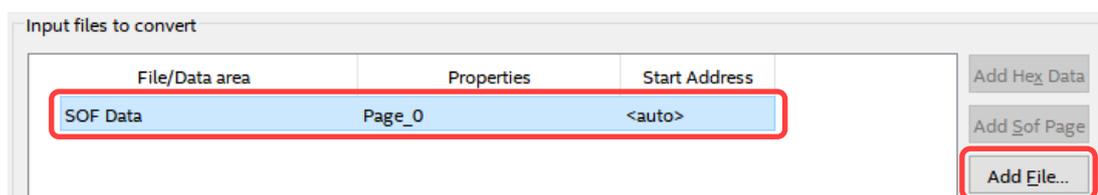
Select Devices ダイアログボックスから、開発ボードに応じたデバイスファミリー (*Device family*) とデバイス名 (*Device name*) を選択し、OK ボタンをクリックします。



## ③ SOF Data 行をハイライト選択し、Add File ボタンをクリックします。

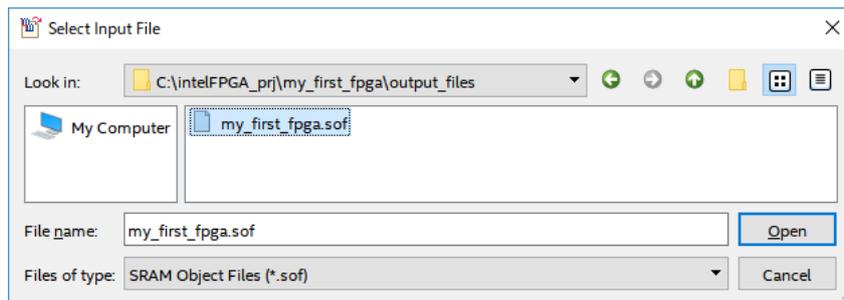


▲ Cyclone® 10 LP Evaluation Kit ユーザーの画面

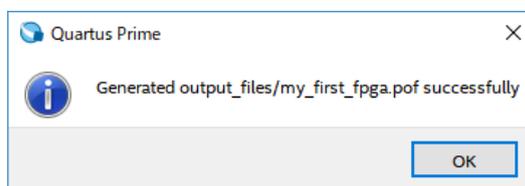
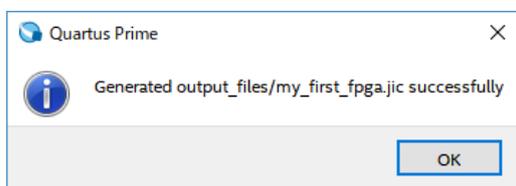


▲ MAX® 10 FPGA Evaluation Kit ユーザーの画面

- ④ `my_first_fpga.sof` ファイルを指定し、**Open** ボタンをクリックします。



4. Convert Programming Files ダイアログボックスの **Generate** ボタンをクリックし、コンフィグレーション ROM 用のプログラミング・ファイルを生成します。
5. Convert Programming Files ダイアログボックスの **Close** ボタンをクリックし、ダイアログボックスを閉じます。



以上で コンフィグレーション ROM 用のプログラミング・ファイルが完成しました。

## 7-2-2. コンフィグレーション ROM へプログラミングする

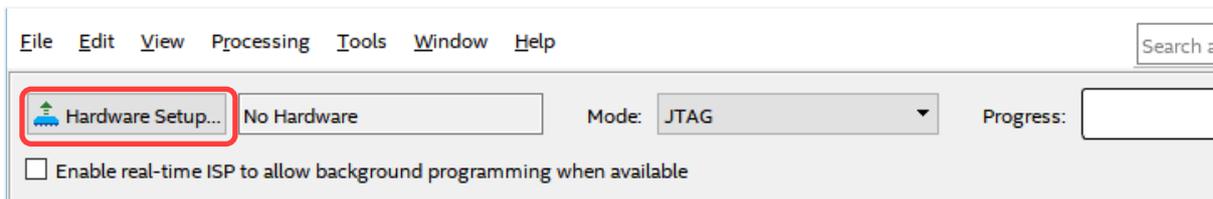
1. ダウンロード・ケーブル、パソコン、評価ボードを接続します。

ボード上の JTAG コネクタの位置や向きは、各評価ボードのユーザーガイドや回路図をご確認ください。

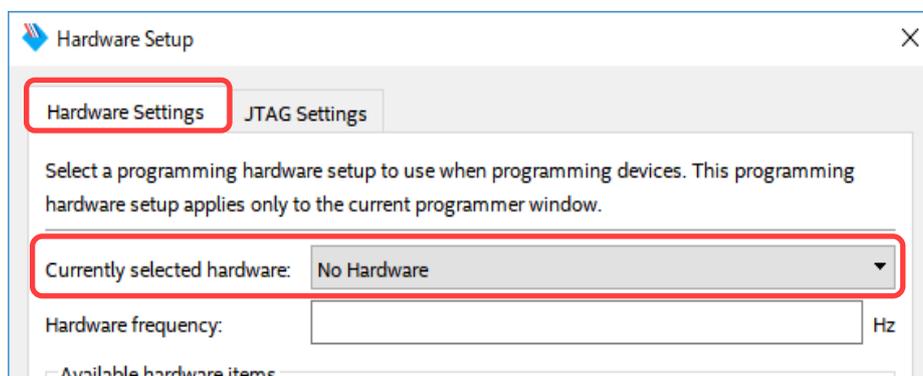
【表 7-2-2-1】ダウンロード・ケーブルの接続

評価ボード	説明
Cyclone® 10 LP Evaluation Kit	<ol style="list-style-type: none"> <li>1. Cyclone® 10 LP Evaluation Kit に付属しているケーブルにおいて、mini USB (Type-B) を評価ボードに接続します。</li> <li>2. USB ケーブルのうち黒いコネクタをパソコンへ接続してください。(ダウンロード・ケーブルと電源供給ケーブルを兼用しています。)</li> <li>3. ボードに電源が供給されます。</li> </ol>
MAX® 10 FPGA Evaluation Kit	<ol style="list-style-type: none"> <li>1. インテル FPGA ダウンロード・ケーブル II 本体と USB ケーブルを接続します。</li> <li>2. インテル FPGA ダウンロード・ケーブル II 本体から出ているケーブルの 10 pin ヘッダーを評価ボード上の JTAG コネクタに接続します。</li> <li>3. 電源用 USB ケーブルで、パソコンと評価ボードを接続します。(ボードに電源が供給されます。)</li> </ol>

2. Quartus® Prime のメニューから **Tools** ➤ **Programmer** を選択し、Programmer ウィンドウを起動します。
3. Programmer ウィンドウの **Hardware Setup** ボタンをクリックします。



4. **Hardware Settings** タブの **Currently selected hardware** 項のプルダウン・リストから、使用するダウンロード・ケーブルを選択し、**Close** ボタンをクリックします。



【表 7-2-2-2】 Currently selected hardware に選択するケーブル名

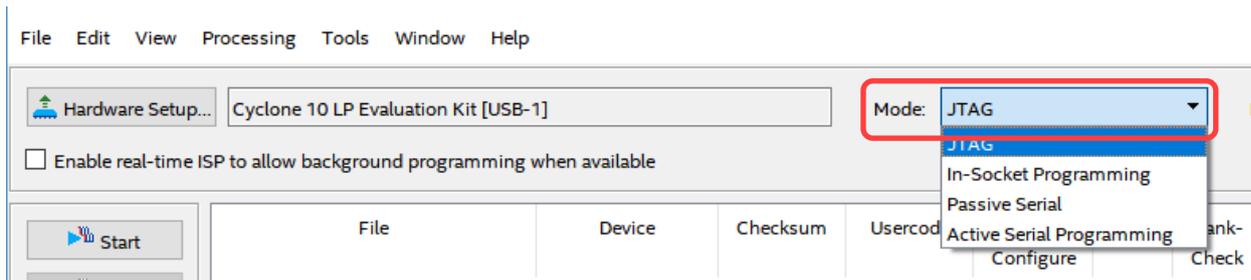
評価ボード	Currently selected hardware に検出される名前
Cyclone® 10 LP Evaluation Kit	Cyclone® 10 LP Evaluation Kit
MAX® 10 FPGA Evaluation Kit	使用するダウンロード・ケーブル名 (例: USB-Blaster II)

## ① Note:

- Cyclone® 10 LP Evaluation Kit は、オンボード インテル FPGA ダウンロード・ケーブル II 対応基板です。よって、ご利用前に [インテル FPGA ダウンロード・ケーブル II のドライバをインストール](#) してください。
- インテル FPGA ダウンロード・ケーブル II は、TCK 周波数を変更可能です。詳細は、下記 Web ページをご確認ください。

[USB-Blaster™ II の TCK 周波数を変更してみよう](#)

5. Programmer ウィンドウの **Mode** プルダウン・リストより、**JTAG** を選択します。



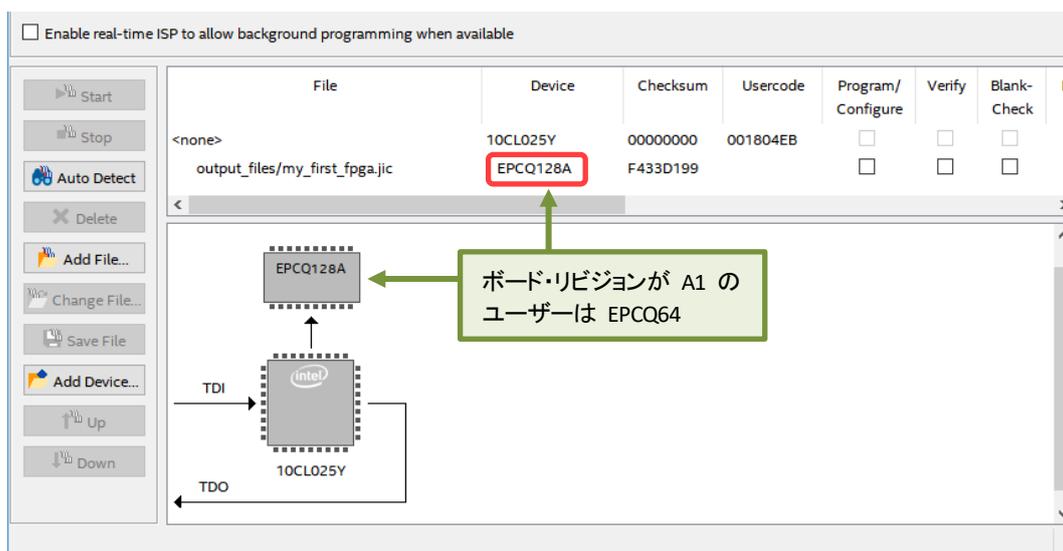
▲ 例: Cyclone® 10 LP Evaluation Kit の場合

6. Programmer ウィンドウの **File** 欄に **my\_first\_fpga.sof** ファイルがすでに登録されている場合には、sof ファイルをハイライト選択し **Delete** ボタンをクリックして削除 (空欄に) してください。
7. Programmer ウィンドウの **Add File** ボタンをクリックし、¥作業フォルダ¥output\_files 内にある **my\_first\_fpga.jic** あるいは **my\_first\_fpga.pof** ファイルを選択し、Open ボタンをクリックして File 欄に登録します。

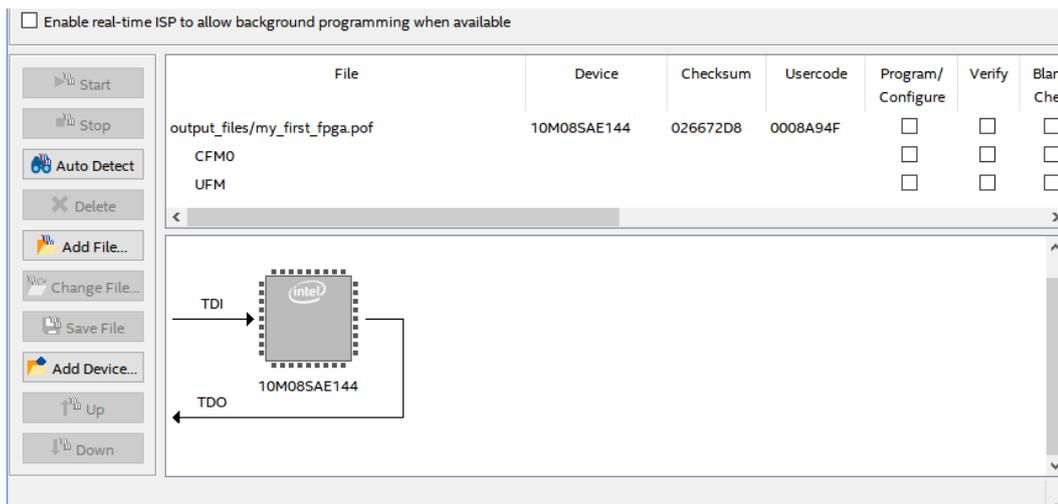
【表 7-2-2-3】 Currently selected hardware に選択するケーブル名

評価ボード	選択するファイル
Cyclone® 10 LP Evaluation Kit	my_first_fpga.jic
MAX® 10 FPGA Evaluation Kit	my_first_fpga.pof

ファイル登録後の Programmer 画面は、このようになります。



▲ Cyclone® 10 LP Evaluation Kit / EPCQ128A の場合



▲ MAX® 10 FPGA Evaluation Kit の場合

8. Programmer ウィンドウのプログラミング・オプションを設定します。

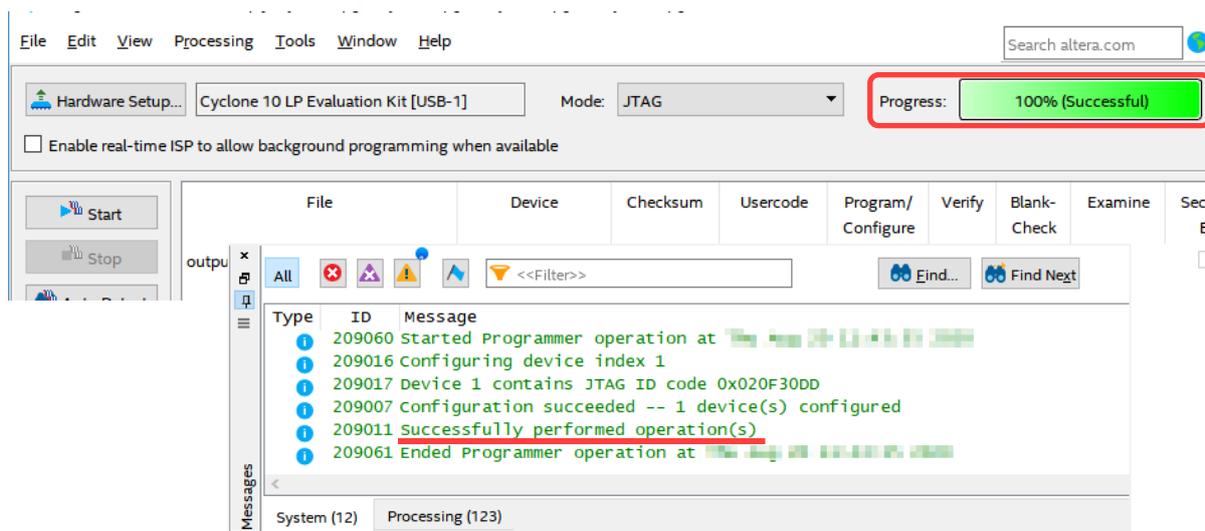
コンフィギュレーション ROM にプログラミングするため、[表 7-2-2-4] を参考に Program/Configure オプションにチェックを入れます。

【表 7-2-2-4】 Program/Configure オプション

評価ボード	Program/Configure オプション																								
Cyclone® 10 LP Evaluation Kit	<input checked="" type="checkbox"/> Factory default enhanced SFL image <input checked="" type="checkbox"/> output_files/my_first_fpga.jic <table border="1"> <thead> <tr> <th>File</th> <th>Device</th> <th>Checksum</th> <th>Usercode</th> <th>Program/Configure</th> <th>Verify</th> </tr> </thead> <tbody> <tr> <td>Factory default enhanced SFL image</td> <td>10CL025Y</td> <td>001804EB</td> <td>001804EB</td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> </tr> <tr> <td>output_files/my_first_fpga.jic</td> <td>EPCQ128A</td> <td>F433D199</td> <td></td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> </tr> </tbody> </table> <p>※ jic ファイル側のオプションを <input checked="" type="checkbox"/> すると、Factory default enhanced SFL image 側も自動的に有効になります。</p>	File	Device	Checksum	Usercode	Program/Configure	Verify	Factory default enhanced SFL image	10CL025Y	001804EB	001804EB	<input checked="" type="checkbox"/>	<input type="checkbox"/>	output_files/my_first_fpga.jic	EPCQ128A	F433D199		<input checked="" type="checkbox"/>	<input type="checkbox"/>						
File	Device	Checksum	Usercode	Program/Configure	Verify																				
Factory default enhanced SFL image	10CL025Y	001804EB	001804EB	<input checked="" type="checkbox"/>	<input type="checkbox"/>																				
output_files/my_first_fpga.jic	EPCQ128A	F433D199		<input checked="" type="checkbox"/>	<input type="checkbox"/>																				
MAX® 10 FPGA Evaluation Kit	<input checked="" type="checkbox"/> output_files/my_first_fpga.pof <input checked="" type="checkbox"/> CFM0 <input type="checkbox"/> UFM <table border="1"> <thead> <tr> <th>File</th> <th>Device</th> <th>Checksum</th> <th>Usercode</th> <th>Program/Configure</th> <th>Verify</th> </tr> </thead> <tbody> <tr> <td>output_files/output_file.pof</td> <td>10M08SAE144</td> <td>04319F46</td> <td>0008B13B</td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> </tr> <tr> <td>CFM0</td> <td></td> <td></td> <td></td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> </tr> <tr> <td>UFM</td> <td></td> <td></td> <td></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> </tr> </tbody> </table>	File	Device	Checksum	Usercode	Program/Configure	Verify	output_files/output_file.pof	10M08SAE144	04319F46	0008B13B	<input checked="" type="checkbox"/>	<input type="checkbox"/>	CFM0				<input checked="" type="checkbox"/>	<input type="checkbox"/>	UFM				<input type="checkbox"/>	<input type="checkbox"/>
File	Device	Checksum	Usercode	Program/Configure	Verify																				
output_files/output_file.pof	10M08SAE144	04319F46	0008B13B	<input checked="" type="checkbox"/>	<input type="checkbox"/>																				
CFM0				<input checked="" type="checkbox"/>	<input type="checkbox"/>																				
UFM				<input type="checkbox"/>	<input type="checkbox"/>																				

9. Programmer ウィンドウの **Start** ボタンをクリックし、コンフィグレーション ROM へのダウンロードを開始します。Progress バーが 100% になり、Quartus Prime の Messages ウィンドウ (System タブ) に Info: Successfully performed operation(s) のメッセージを確認できたら、データのダウンロードは完了です。

▼ Programmer ウィンドウ



▲ Messages ウィンドウ

📖 参考:

- 以下に、ここでの作業に関連したドキュメントおよびコンテンツをご案内します。ご参考ください。
  - [「Quartus Prime はじめてガイド - デバイス・プログラミングの方法」](#)
  - [「Quartus Prime - プログラミング・ファイルの生成と変換 \(Convert Programming Files\)」](#)
  - [FPGA 経由で EPCQ デバイスへプログラミング \(JIC プログラミング\)](#)

以上で コンフィグレーション ROM への書き込みは完了です。

評価ボード上の 再コンフィグレーション用ボタンを押下し、sof ファイルを書き込んだ際と同様の動作が行われること (8章 を参照) を確認してください。

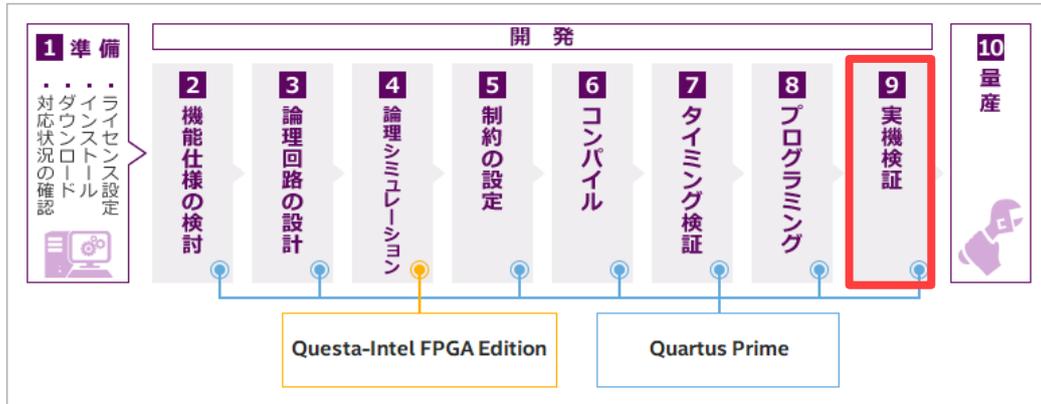
各評価ボードの再コンフィグレーション用ボタンは、以下のとおりです。

【表 7-2-2-5】再コンフィグレーション用ボタン

評価ボード	Cyclone® 10 LP Evaluation Kit	MAX® 10 FPGA Evaluation Kit
シルク印刷番号	S1	SW2

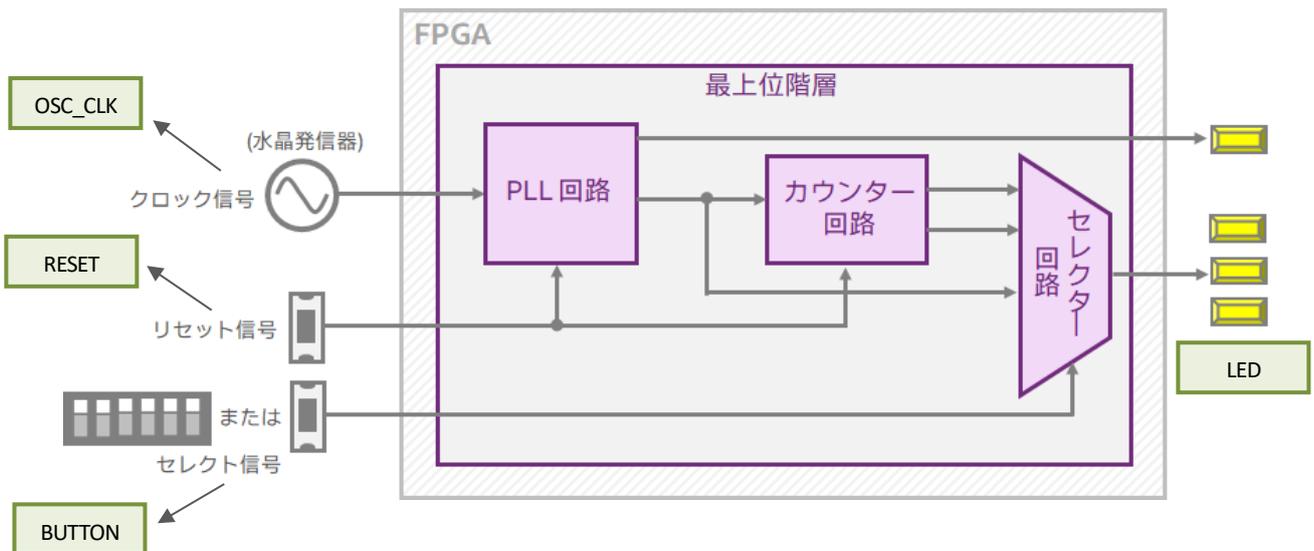
## 8. 実機で動作検証をする

実機（評価ボード）上で FPGA が正しく動作するかを確認します。（この作業は、開発フローの [9] にあたります。）



このチュートリアルでは、LED が点灯および消灯する（いわゆる L チカ）動作を作成しました。

すでに開発ボード上の LED が点灯しているはずですが、セレクト信号を割り当てたピンに接続されているタクトスイッチを押す（Cyclone® 10 LP Evaluation Kit ユーザー）、あるいは DIP スイッチを切り替える（MAX® 10 FPGA Evaluation Kit ユーザー）と、点灯する間隔が変わります。



### <FPGA の動作仕様>

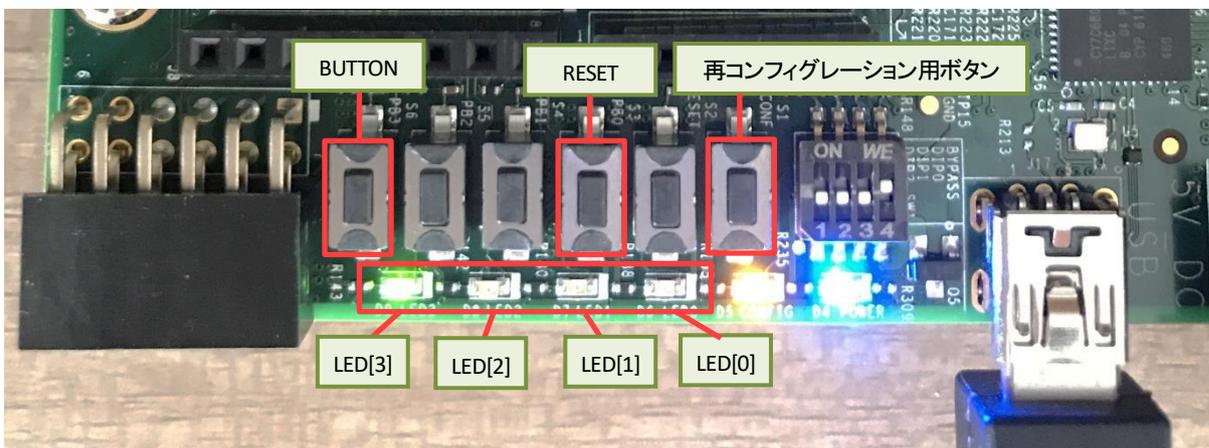
- LED が点灯および点滅します。
- BUTTON に割り当てたピンに接続されているタクトスイッチを押下している間 または DIP スイッチが On 側に位置する場合は、LED の点滅する速さが変わります(遅くなります)。
- RESET に割り当てたピンに接続されているタクトスイッチを押下している間、PLL 回路とカウンター回路がリセットされ、LED が全消灯します。

## L チカの動作は成功しましたか？

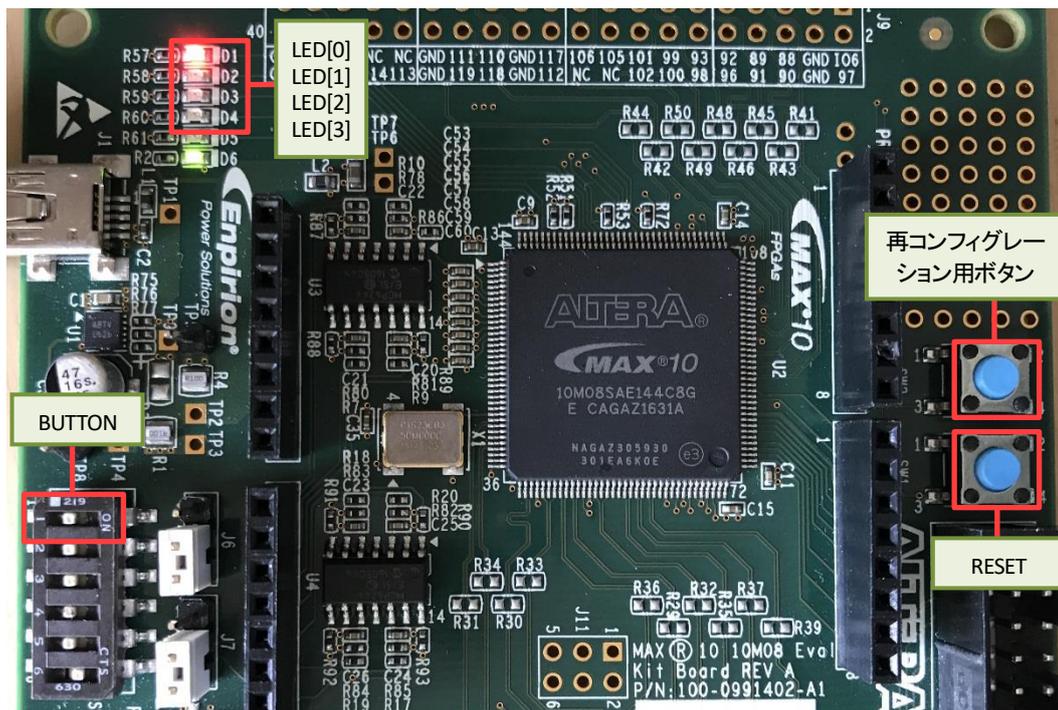
実機動作	次の作業ステップ						
sof ファイルの書き込み: 問題なし	コンフィグレーション ROM へプロジェクトのデータを書き込みます。 (7-2 章へ)						
sof ファイルの書き込み: 問題あり	<p>デザインコードやピンアサインを見直してください。</p> <p>↓</p> <table border="1"> <thead> <tr> <th>修正点</th> <th>修正後に実施する作業</th> </tr> </thead> <tbody> <tr> <td>デザインコード</td> <td> <ul style="list-style-type: none"> <li>論理シミュレーション (3 章へ)</li> <li>コンパイル (5 章へ)</li> </ul> </td> </tr> <tr> <td>ピンアサイン</td> <td> <ul style="list-style-type: none"> <li>コンパイル (5 章へ)</li> </ul> </td> </tr> </tbody> </table> <p>↓</p> <p>再コンパイルで生成した sof ファイルを使用し、Programmer でダウンロード。(7-1 章へ)</p> <p>↓</p> <p>実機検証を行い、動作を確認します。(8 章へ)</p>	修正点	修正後に実施する作業	デザインコード	<ul style="list-style-type: none"> <li>論理シミュレーション (3 章へ)</li> <li>コンパイル (5 章へ)</li> </ul>	ピンアサイン	<ul style="list-style-type: none"> <li>コンパイル (5 章へ)</li> </ul>
修正点	修正後に実施する作業						
デザインコード	<ul style="list-style-type: none"> <li>論理シミュレーション (3 章へ)</li> <li>コンパイル (5 章へ)</li> </ul>						
ピンアサイン	<ul style="list-style-type: none"> <li>コンパイル (5 章へ)</li> </ul>						

各評価ボードの LED、タクトスイッチ、DIP スイッチの位置は、次のとおりです。

▼ Cyclone® 10 LP Evaluation Kit の場合



▼ MAX® 10 FPGA Evaluation Kit の場合



以上で 実機での動作検証は完了です。評価ボードの電源を OFF にし、その後 ダウンロードケーブルを脱着してください。

以上で このチュートリアルの作業はすべて終了です。

#### 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。  
[株式会社マクニカ 半導体事業 お問い合わせフォーム](#)
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。