

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



Quartus Prime はじめてガイド Quartus Prime 簡易チュートリアル

<u>目次</u>

| 1. | は | じめに | 3 |
|----|-----------------------------------|---|--|
| | 1-1. | 必要な環境 | . 3 |
| | 1-2. | サンプル・デザインの概要 | . 4 |
| 2. | Qı | Jartus Prime の設計 | 5 |
| | 2-1. | 論理回路を設計する | . 5 |
| | | 2-1-1. プロジェクトを作成する | . 6 |
| | | 2-1-2. 論理回路を作成する | . 9 |
| | | 2-1-3. Analysis and Elaboration の実行 | 21 |
| | 2-2. | 論理シミュレーションを実行する | 22 |
| | 2-3. | 制約の設定をする | 27 |
| | | 2-3-1. ユーザ 1/0 ピンのアサイン | 28 |
| | | 2-3-2. タイミング制約 | 30 |
| | | | |
| | 2-4. | コンパイルを実行する | 31 |
| | 2-4. 2-5. | コンパイルを実行する タイミングを検証する | 31 32 |
| | 2-4. 2-5. 2-6. | コンパイルを実行する タイミングを検証する プログラミング | 31 32 33 |
| | 2-4. 2-5. 2-6. | コンパイルを実行する タイミングを検証する プログラミング 2-6-1. sof ファイルを書き込む | 31 32 33 34 |
| | 2-4. 2-5. 2-6. | コンパイルを実行する タイミングを検証する プログラミング 2-6-1. sof ファイルを書き込む 2-6-2. コンフィギュレーション・デバイスに書き込む | 31 32 33 34 37 |
| | 2-4. 2-5. 2-6. | コンパイルを実行する タイミングを検証する プログラミング 2-6-1. sof ファイルを書き込む 2-6-2. コンフィギュレーション・デバイスに書き込む 2-6-2-1. プログラミング・ファイルを作成する | 31 32 33 34 37 37 |
| | 2-4. 2-5. 2-6. | コンパイルを実行する タイミングを検証する プログラミング 2-6-1. sof ファイルを書き込む 2-6-2. コンフィギュレーション・デバイスに書き込む 2-6-2-1. プログラミング・ファイルを作成する | 31 32 33 34 37 37 39 |
| | 2-4. 2-5. 2-6. 2-7. | コンパイルを実行する | 31 32 33 34 37 37 39 43 |
| 3. | 2-4. 2-5. 2-6. 2-7. ス | コンパイルを実行する タイミングを検証する プログラミング | 31 32 33 34 37 37 39 43 45 |

1. <u>はじめに</u>

この資料は、インテル[®] Quartus[®] Prime 開発ソフトウェアを使用して、デジタル論理回路の作成からインテル FPGA の搭載された評価キットや開発キット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。初めて FPGA 設計をするユーザが、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。

- Notes: ・ 本チュートリアルは Quartus[®] Prime 18.0 以降の環境でご利用になることを推奨します。同様に、 ModelSim[®] - Intel[®] FPGA Edition (以下、ModelSim - Intel FPGA Starter Edition を含む)は、ご利用になる Quartus Prime バージョンと同時提供されているバージョンをご利用ください。
 - ・Quartus Prime および ModelSim Intel FPGA Edition は、タイプ(Edition)により有償のライセンスが必 要です。詳細は下記ページをご覧ください。

ほんとのほんとの導入編 その2. FPGA 開発をはじめるために準備するもの

下図は、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 ModelSim - 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]~[8] 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。

FPGA / CPLD の一般的な開発フロー/トップページ

1-1. 必要な環境

本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。

- Quartus Prime (Lite あるいは Standard Edition のいずれか) 18.0 以降、および ModelSim Intel FPGA
 Edition をインストールしたパソコン
 - 使用可能な Windows オペレーティング・システムは、下記の対応表でご確認ください。
 <u>「Quartus Prime サポート Windows OS 対応表」</u>



Note: ・ Linux のディストリビューションについては、以下のページをご覧ください。

Operating System Support

- 各種ソフトウェアのダウンロードについてはこちら、インストールについてはこちらをご覧ください。
- インテル FPGA ダウンロード・ケーブル II (旧 USB-Blaster™II)※
 - インテル FPGA ダウンロード・ケーブル(旧 USB-Blaster)[※]、または インテ ル FPGA イーサネット・ケーブル(旧 EthernetBlaster II)でも可
 - Notes: ・ご利用になるインテル製の FPGA 評価キットや開発キットにより、上記ダ ウンロード・ケーブルと同等の機能が内蔵されているボード(例: On-Board USB Blaster II 搭載タイプ)もあります。その場合は、これらのダウンロード・ ケーブルは不要です。



・上記のダウンロード・ケーブルおよびオンボード・タイプをパソコンではじめて利用するには、専 用ドライバのインストールが必要です。インストールの手順は、下記 TIPS をご参考ください。

【TIPS】 USB - Blaster ™ II のドライバをインストールしてみよう 【TIPS】 USB - Blaster ™ のドライバをインストールしてみよう

- インテル FPGA の搭載された開発キット
 - チュートリアルの対象ボードは下表のとおりです

| Board | Device Family | Available devices | ダウンロード・ケーブル |
|---|-----------------------------------|-------------------|-------------------------------|
| MAX [®] 10 FPGA Evaluation Kit | MAX 10 | 10M08SAE144C8G | 必要 |
| Cyclone [®] 10 LP Evaluation Kit | 0 LP Evaluation Kit Cyclone 10 LP | | 不要 On-Board USB-Blaster II |
| Beryll Cyclone V GX Base Board Cyclone V | | 5CGXFC4C6F27C7 | 不要 On-Board USB-Blaster |

1-2. サンプル・デザインの概要

このチュートリアルでは、デジタル論理回路をハードウェア記述言語(以下 HDL)で設計します。

開発ボード上の LED を、入力ボタンにより点滅する速度を制御するの論理回路を作成します。ブロック図は 以下のとおりです。

この論理回路は、3つの機能で構成します。





• 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逓倍や分周、およびクロックの位相をシフトすることが可能な 回路です。

・カウンタ回路

入力されたクロックの数を数えて処理を行う回路です。

・セレクタ回路(マルチプレクサ回路)

複数の入力のうちのどれか1つを選んで取り出す回路です。

Note: ・ HDL 設計の経験がないユーザは、以下もご活用ください。

<u>はじめてみよう! Verilog-HDL <演習問題つき></u>

<u>はじめてみよう! VHDL <演習問題つき></u>

2. <u>Quartus Prime の設計</u>

Quartus Prime を起動します。

Windows OS の場合、インストール時に作成したデスクトップ上のショートカット・アイコンをダブルクリック、も

しくは[スタート]メニュー (Windows 8.1 の場合は スタート画面)から Quartus Prime 開発ソフトウェア を起動させます。

Linux OS の場合には、quartus コマンドにより 起動させます。



2-1. 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 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倫理回路を作成し、登録を行います。

2-1-1. プロジェクトを作成する

論理回路を作成するにあたり、プロジェクトを作成します。

1. File メニュー ➤ 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。

| File | Edit View Project | t Assignments P | rocessing | |
|------|--------------------|---------------------------------------|---|---|
| | New | Ctrl+N | | |
| a | Open | Ctrl+O | x 🐟 | |
| | Close | Ctrl+F4 | 8 × | |
| A | New Project Wizard | | | |
| £ | Open Project | Ctrl+J | | |
| | Save Project | | | |
| | Close Project | | | |
| B | Save | 🕞 New Project Wizard | | |
| 73.5 | Save As | Introduction | | |
| 嗝 | Save All | | | |
| | | The New Project W | izard helps you create a new project and preliminary project settings, including the following: | |
| | | Proj Nan | ect name and directory ne of the top-level design entity | |
| | | • Proj | ect files and libraries | |
| | | Targ EDA | zet device family and device | |
| | | You can change th | e settings for an existing project and specify additional project-wide settings with the Settings | |
| | | command (Assign | ments menu). You can use the various pages of the Settings dialog box to add functionality to the | |
| | | project. | | |
| | | | | |
| | | | Introduction ウィンドウを次回以降表示させない 提合は、このナポションにエーックを入れます | |
| | | Don't show me | はhis introduction again | |
| | | | 0 | |
| | | | <back next=""> Finish Cancel Heln</back> | R |
| | | | There for the server first | 4 |

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。

| Directory, Name, Top-Level Entity ページ | | | | |
|---|--------------------------------|--|--|--|
| What is the working directory for this project? | C:¥intelFPGA_prj¥my_first_fpga | | | |
| What is the name of this project? | my_first_fpga | | | |
| What is the name of the top-level design entity for this project? This name is (省略) | my_first_fpga | | | |
| | | | | |

Note: ・ 作業ディレクトリのパス、プロジェクト名、ファイル名は、半角・英数字のみご利用ください。全角 やスペースの使用はできません。-

| C:/intelFPGA pri/my first fp | ioi uns project: | | |
|---|------------------|--|--|
| What is the name of this proj | ect? | | |
| my_first_fpga | | | |
| What is the name of the top-level design entity for this project? This name is case sensitive ar exactly match the entity name in the design file. | | | |
| my_first_fpga | | | |
| Use Existing Project Settings | | | |
| | | | |

パソコンにあらかじめ C:¥intelFPGA_prj¥my_first_fpga ディレクトリを作成していない場合は、入力した ディレクトリの作成を促すメッセージ(下図)が表示されます。Yes を選択し、ワーキング・ディレクトリを作 成します。

| 🕤 Quar | rtus Prime | | <u></u> |
|--------|--|-------------------------------------|----------------|
| | Directory "C:/intelFPGA_prj/my_ it? | _first_fpga" does not exist. Do you | want to create |
| | | Yes | No |

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 ページは、何も選択せずに Next ボタンをクリックします。
- 5. Family, Device & Board Settings ページでは、使用する開発キットに搭載されているインテル FPGA の型 番を選択します。

| Family, Device & Board Settings ページ | | | | | | |
|---|---------------|---------|--------------|------------------------|-------------------|--|
| Board Device Family | | Package | Pin count | Core speed grade | Available devices | |
| MAX [®] 10 FPGA Evaluation Kit | MAX 10 | EQFP | 144 | 8 | 10M08SAE144C8G | |
| Cyclone [®] 10 LP Evaluation Kit | Cyclone 10 LP | UFBGA | 256 | 7 | 10CL025YU256I7G | |
| Beryll Cyclone V GX Base Board | Cyclone V | FPGA | 672 | 7_H6 | 5CGXFC4C6F27C7 | |

- ① Family(Device family エリア内)プルダウン・リストから、FPGA の<u>デバイス・ファミリ</u>を選択します。
- ② Show in 'Available devices' list エリアにおいて、上表を参考に <u>Package</u>、<u>Pin count</u>、<u>Core speed grade</u> の項目を選択します。選択した項目がフィルタ条件となり、Available devices 欄に該当する型番のみ が表示されます。
- ③ Available devices 欄から対象の型番をクリックで指定し、ハイライトさせます。
 - Note: ・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。

| - 1 C | Board | | | | | | |
|---|--|--|--|---|--|-------------------------------|--|
| Select the f You can ins To determi | amily and o stall addition ne the vers | device you want to t mal device support ion of the Quartus F | target for co with the In: Prime softw | ompilation. stall Devices com vare in which you | mand on the Tool r target device is s | s menu. upported, refer to |) the <u>Device Support List</u> webpage |
| Device family | | | | | Show in 'Availat | ole devices' list | |
| Eamily: Cyclone 10 LP | | | | • | Package: | UFBGA | • |
| Device | : All | | | ¥ | Pin <u>c</u> ount: | 256 | • |
| Target de | vice | | | | Core sp <u>e</u> ed grad | de: 7 | • |
| Auto | device sele | cted by the Fitter | | | Name filter: 025 | | |
| Speci O Other | fic device s | elected in 'Available | e devices' li | st | Show advanced devices | | |
| A <u>v</u> ailable d | evices: | | | | | | |
| Na | me | Core Voltage | LEs | Total I/Os | GPIOs | Memory Bits | Embedded multiplier 9-bit |
| 10CL025Y | ′U256A7G | 1.2V | 24624 | 151 | 151 6 | 08256 | 132 |
| 10CL025Y | U25617G | 1.2V | 24624 | 151 | 151 6 | 08256 | 132 |
| | | | | | | | |



6. Finish ボタンをクリックします。

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ることを確認します。



関連情報: 下記ドキュメントをご参考ください。

<u>「Quartus Prime はじめてガイド - プロジェクトの作成方法」</u>

2-1-2. 論理回路を作成する

作成したプロジェクトに、1-2. サンプル・デザインの概要 で示した論理回路(以下、デザイン)を作成します。

このチュートリアルでは、カウンタ回路およびセレクタ回路は VHDL でコードを記述し、PLL 回路は IP Catalog と言う入力手法で設計します。また、この 3 つのモジュールを接続する最上位階層デザインは、VHDL で記述します。

■ カウンタ回路を設計する

VHDL で作成します。

File メニュー ➤ New ➤ VHDL File (Design Files 内)をダブルクリックで選択します。

Vhdl1.vhd の新規シートが開きます。

| New Quartus Prime Project | |
|------------------------------|--|
| Design Files | |
| AHDL File | |
| Block Diagram/Schematic File | |
| EDIF File | |
| Qsys System File | |
| State Machine File | |
| SystemVerilog HDL File | |
| Tcl Script File | |
| Verilog HDL File | |
| VHDL File | |



- 2. File メニュー ➤ Save As... により、これから作成するデザイン・ファイルに名前を付け、保存するディレクトリを指定します。
 - ・ 保存するディレクトリ: C:¥intelFPGA_prj¥my_first_fpga
 - ・ ファイル名: simple_counter
 - ・ファイルの種類: VHDL Files (*.vhd *.vhdl)
 - ・ Add file to current project オプション = On

| 🗸 🗸 🔻 🚺 🗸 intelFPGA_prj | ▶ my_first_fpga ▶ 		 ▼ 😽 my_ | first_fpgaØ検索 🛛 🔎 |
|--------------------------------|------------------------------------|-------------------|
| 整理 ▼ 新しいフォルダー | | 88 - 0 |
| i my_first_fpga | db | |
| ファイル名(N): simple_co | unter | ÷ |
| ファイルの種類(<u>T</u>): VHDL File | s (*.vhd *.vhdl) | |
| ▲ フォルダーの非表示 | ☑ Add file to current project 保 | 存(S) キャンセル |

3. 次の HDL コードを simple_counter.vhd (または ファイルに入力します。

| library ieee; use ieee.std_logic_1164.all; use ieee.numeric_std.all; | | | | | |
|--|-----------------------------------|---|--|--|--|
| entity sim | nple_counter is | S | | | |
| port | | | | | |
| (| | | | | |
| C | clock | : in std_logic; | | | |
| r | reset | : in std_logic; | | | |
| C | counter_out | : out unsigned (31 downto 0) | | | |
|); | | | | | |
| end entit | :у; | | | | |
| architectu signal o | ure rtl of simpl cnt : unsigne | e_counter is d (31 downto 0) := (others => '0'); | | | |

begin process (clock, reset) begin if (reset = '0') then $cnt \ll (others \implies '0');$ elsif (rising_edge(clock)) then $cnt \leq cnt + 1;$ end if; counter_out <= cnt; end process; end rtl; ٠ simple counter.vhd -🔲 📅 🐺 📳 🗗 🎦 🖞 🔽 💯 🚟 library ieee; use ieee.std_logic_1164.all; use ieee.numeric_std.all; 1 23 4 5 □entity simple_counter is port (6789 Ġ clock : in std_logic; reset : in std_logic; counter_out : out unsigned (31 downto 0) 10 end entity; 11 12 13 □architecture rtl of simple_counter is
_signal cnt : unsigned (31 downto 0) := (others => '0'); 14 15 16 ⊡begin process (clock, reset) begin 18 19 20 21 22 23 24 25 ē if (reset = '0') then cnt <= (others => '0'); elsif (rising_edge(clock)) then cnt <= cnt + 1; end if; counter_out <= cnt;</pre> end process; end rtl; 26

- File メニュー ➤ Save (または Ctrl + S) により simple_counter.vhd を保存します。 4.
- simple_counter.vhd のタブの端にある Close ボタンをクリックし、ファイルを閉じます。 5.



27 20

■ セレクタ回路を設計する

VHDL で作成します。

- 1. File メニュー ➤ New ➤ VHDL File (Design Files 内)をダブルクリックで選択します。
- 2. File メニュー ➤ Save As... により、これから作成するデザイン・ファイルに名前を付け、保存するディ レクトリを指定します。
 - ・ 保存するディレクトリ: C:¥intelFPGA_prj¥my_first_fpga
 - ・ ファイル名: counter_bus_mux
 - ・ ファイルの種類: VHDL Files (*.vhd *.vhdl)
 - ・ Add file to current project オプション = On
- 3. 次の VHDL コードを counter_bus_mux.vhd ファイルに入力します。

```
library ieee;
use ieee.std logic 1164.all;
use ieee.numeric_std.all;
entity counter_bus_mux is
   port
   (
                   : in unsigned (3 downto 0);
        dataa
                   : in unsigned (3 downto 0);
        datab
        sel
                   : in std_logic;
        result
                   : out unsigned (3 downto 0)
   );
end entity;
architecture rtl of counter_bus_mux is
begin
   process (dataa, datab, sel)
   begin
      case (sel) is
        when '1' => result <= dataa;
        when '0' => result <= datab;
        when others => result <= (others => '0');
      end case;
   end process;
end rtl;
```

- 4. File メニュー ➤ Save (または Ctrl + S) により counter_bus_mux.vhd を保存します。
- 5. counter_bus_mux.vhd のタブの端にある Close ボタンをクリックし、ファイルを閉じます。

■ PLL 回路を設計する

IP Catalog で作成します。

- Note: ・ IP Catalog とは、インテル FPGA の IP コア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、<u>About the IP Catalog and Parameter Editor</u> をご参考ください。
- 1. Tools メニュー ➤ IP Catalog を選択し、IP Catalog ウィンドウを表示します。



2. 検索フィールドに *pll* と PLL IP の一部を入力すると、対象 FPGA ファミリで使用できる IPコア名が検索されます。

| IP Catalog | 0 5 × |
|---|--------------------|
| 🔍 pil 🔍 | × = |
| Installed IP Library | ここへ探したい IP 名の一部を入力 |
| Basic Functions Clocks; PLLs and Resets ALTCLKCTRL Intel FPGA IP PLL | 該当する IP だけが検出 |
| ALTPLL ALTPLL_RECONFIG Search for Partner IP | |
| + Add | |



このチュートリアルでは、使用する開発キットにより IP コアが異なります。下表を参考にしてください。

| Board | Device Family | PLL 回路の IP 名 |
|--------------------------------|---------------|-------------------|
| MAX 10 FPGA Evaluation Kit | MAX 10 | ALTPLL |
| Cyclone 10 LP Evaluation Kit | Cyclone 10 LP | ALTPLL |
| Beryll Cyclone V GX Base Board | Cyclone V | PLL Intel FPGA IP |

Note: ・ デバイス・ファミリにより選択できる PLL IP 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. 該当する PLL IP をハイライト選択し、+ Add ボタンをクリック(または IP コアをダブルクリックでも可) すると、Save IP Variation ダイアログボックスが表示されます。

下記の通り、これから作成する PLL IP の保存ディレクトリ、およびモジュール名を入力し、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。

- IP variation file name: C:¥intelFPGA_prj¥my_first_fpga/pll
- IP variation file type : VHDL

| P variation file | name: | |
|------------------|-----------------------|--------------|
| C:/intelEPCA | pri/my first fogs/oll | ОК |
| C./III.CEFEGA_ | pri/my_mst_rpga/pit | Cancel |
| IP variation fi | le type | |
| VHDL | 作成する IP のモジュール名 | を末尾に入り |
| | | 2 () |

OK ボタンをクリックします。

- 4. PLL IP のパラメータを設定します。PLL のタイプにより項目が異なりますので、使用するボードに応じ て設定を行ってください。
 - ALTPLL (MAX 10 FPGA Evaluation Kit / Cyclone 10 LP Evaluation Kit) の設定
 - PLL Intel FPGA IP (Beryll Cyclone V GX Base Board)の設定



- ALTPLL (MAX 10 FPGA Evaluation Kit / Cyclone 10 LP Evaluation Kit)の設定
 - MegaWizard Plug-In Manager ウィンドウの 1 Parameter Settings ➤ General/Modes において、以下の設定をします。
 - a. Currently selected device family に使用するボードのデバイス・ファミリが表示されている ことを確認します。
 - b. Which device speed grade will you be using? に下表の該当する値を選択します。
 - c. What is the frequency of the inclockO input? に下表の該当する値を入力し、単位を選択し ます。
 - d. Next ボタンをクリックします。

| Parameter Settings > General/Modes | | | | |
|------------------------------------|---|--|--------|--|
| Board | Which device speed grade will you be using? | What is the frequency of the inclock0 input? | その他の設定 | |
| MAX 10 FPGA Evaluation Kit | 8 | 50 MHz | デフォルト | |
| Cyclone 10 LP Evaluation Kit | 7 | 50 MHz | デフォルト | |

| MegaWizard Plug-In Manager (page 1 of 12) | | 2 🔀 |
|---|--|-----------------------------------|
| ALTPLL | | About Documentation |
| PlL Settings Reconfiguration Clocks | A 5Summary | |
| General/Modes Inputs/Look Bandwidtr/JS | Currently selected | d device family: Cyclone 10 LP |
| pii | | Match project/default |
| inclk0 inclk0 frequency: 50.000 MHz c0 operation Mode: Normal locked | Able to implement the requested PLL General | |
| 0 1/1 0.00 00.001 | Which device speed grade will you be using? | 7 |
| Cyclone 10 LP | Use military temperature range devices only | |
| | What is the frequency of the inclk0 input? | 50.000 MHz 💌 |
| | Set up PLL in LVDS mode Data r | rate: Not Available 👻 Mbps |
| | PLL Type Which PLL type will you be using? | Select the PLL type automatically |
| | Operation Mode How will the PLL outputs be generated? (9) Use the feedback path inside the PLL | |
| | In normal mode | |
| | In source-synchronous compensation Mode | |
| | In zero delay buffer mode Connect the Aminis part (hidiractional) | |
| | With no compensation | |
| | Create an 'fbin' input for an external feedback (External F | Feedback Mode) |
| | Which output clock will be compensated for? | [c0 ▼] |
| | G | ancel < Back Next > Finish |



Parameter Settings ➤ Inputs/Lock において、以下のとおりに設定します。

| Parameter Settings ➤ Inputs/Loc | k | | |
|---------------------------------|--|---------------------------|--------|
| Board | Create an 'areset' input to asynchronously reset the PLL | Create 'locked' output | その他の設定 |
| MAX 10 FPGA Evaluation Kit | On | Off | デフォルト |
| Cyclone 10 LP Evaluation Kit | On | Off | デフォルト |

オフにしたポートは PLL ブロックのグラフィカルなプレビューから消えます。

| MegaWizard Plug-In Manager [page 2 of 12] | 8 |
|---|--|
| ALTPLL | About Documentation |
| Parameter PL 3 Output Settings Reconfiguration Clocks | EDA Summary |
| General/Modes Inputs/Lock Band | vidth/SS $>$ Clock switchover $>$ |
| pli | Able to implement the requested PLL |
| jnclk0 ct | Optional Inputs |
| areset Operation Mode: Normal | ✓ Create an 'areset' input to asynchronously reset the PLL |
| Cik Ratio Ph (dg) DC (%) | Create an 'pfdena' input to selectively enable the phase/frequency detector |
| 0.00 000 | Lock Output |
| Cyclone 10 LP | Create 'locked' output |
| | Enable self-reset on loss lock |
| | Advanced Parameters Using these parameters is recommended for advanced users only |
| | - Configurations with output dock(s) that use cascade counters are not supported |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | Cancel < Back Next > Finish |

Next ボタンをクリックします。

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

Clock division factor に下表の該当する値を入力します。(または上下ボタンで数値を設定します。)

| Output Clocks ➤ clk c0 | | |
|------------------------------|--------------------------|--------|
| Board | Clock division factor | その他の設定 |
| MAX 10 FPGA Evaluation Kit | 10 | デフォルト |
| Cyclone 10 LP Evaluation Kit | 10 | デフォルト |

| NegaWizard Plug-In Manager [page 6 of 12] | | | ? |
|--|---|--|-----------------------------|
| | | | About Documentation |
| Parameter PLL 3 Output Settings Reconfiguration Clocks | EDA 5 Summary | | |
| dkc0 dkc1 > dkc2 > dkc3 > | dk c4 > | | |
| pli | c0 - Core/External Output Cloc Able to implement the requested PLL | ck. | |
| inclk0 c0 | Vise this dock | | |
| areset Operation Mode: Normal | Clock top Settings | Requested Settings | Actual Settings |
| Clk Ratio Ph (dg) DC (%) | Enter output dock frequency: | 100,0000000 MHz - | 5.000000 |
| 60 1/10 0.00 50.00 | Enter output clock parameters: Clock multiplication factor | 1 | 1 |
| Cyclone 10 LP | Clock division factor | 10 × << Co | 10 |
| | Clock phase shift | 0.00 두 deg | • 0.00 |
| | Clock duty <mark>cycle</mark> (%) | 50.00 | 50.00 |
| | | Description | Val. |
| | Note: The displayed internal settings of the | Primary clock VCO frequency (| (MHz) 6 |
| | PLL is recommended for use by advanced users only | Modulus for M counter | 12 + |
| | | Per Clock Feasib <mark>c0</mark> c1 c | ility Indicators 2 c3 c4 |

- ④ Finish ボタンをクリックします。
- ⑤ 5 Summary では、生成するファイルを選択できます。

このチュートリアルでは、*.cmp ファイル、*_inst.vhd ファイルを選択します。

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 ALTPLL の作成は完了です。< 19ページへ進みます。>



- PLL Intel FPGA IP (Beryll Cyclone V GX Base Board)の設定
 - ① General タブにおいて、以下の設定をします。

| General タブ | |
|---------------------------|--------|
| Reference Clock Frequency | 50 MHz |
| Enable locked output port | OFF |
| Desired Frequency | 5 MHz |
| Actual Frequency | 5 MHz |
| その他の設定 | デフォルト |



② Finish ボタンをクリックすると、ファイル生成が開始されます。

Generation ダイアログボックスに Generation Successful が表示されたら Exit ボタンをクリックします。以下右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。

| Info: pll: Variation language : VHDL Info: pll: Output directory : C:\intelFPGA_prj\m; | 🔾 Quartus Prime IP Files | |
|---|---|--|
| Info: pil: Generating variation file C:\intelFPGA Info: pil: Generating synthesizable HDL desig Info: Generating altera_pil "pil" for QUARTUS Info: "pil" instantiated altera_pil "pil" | When you create an Intel IP variation, a Quartus Prime IP File is generated. Quartus Prime IP Files are used to represent the Intel IP in your design. Do you want to add the Quartus Prime IP File to the project? | |
| Info: Done pill with 2 modules, 3 files Info: pll: Generating simulation model Info: Generating altera_pil "pill" for SIM_VHDL Info: Generating simgen model | C:\intelFPGA_prj\my_first_fpga\pll.qip | |
| Generating integration files : | Automatically add Quartus Prime IP Files to all projects (Note: Turning on this option permanently suppresses this dialog box. You can change this setting in the Options dialog box) | |





■ 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し、最上位階層のエンティティを作成します。VHDL で回路を完成 させます。

- 1. File メニュー ➤ New ➤ VHDL File (Design Files 内)をダブルクリックで選択します。
- 2. File メニュー ➤ Save As... により、これから作成するデザイン・ファイルに名前を付け、保存するディレクトリを指定します。
 - ・ 保存するディレクトリ: C:¥intelFPGA_prj¥my_first_fpga
 - ファイル名: my_first_fpga
 - ・ ファイルの種類: VHDL Files (*.vhd *.vhdl)
 - ・ Add file to current project オプション = On
- 3. 次の VHDL コードを my_first_fpga.vhd ファイルに入力します。

| library ieee; use ieee.std_logic_1164.all; use ieee.numeric_std.all; | |
|--|---|
| entity my_first_fpga is port (OSC_CLK : in std_logic; RESET : in std_logic; BUTTON : in std_logic; LED : out unsigned (3 downto 0)); end entity; | |
| architecture rtl of my_first_fpga is | |
| signal pll_c0 : std_logic; signal cnt : unsigned (31 downto 0); signal tmp : unsigned (3 downto 0); | PLL Intel FPGA IP デザイン (Beryll Cyclone V GX Base Board) の場合は、左の点線枠内を下記に 変更してください。 |
| <pre>component pll port (areset : in std_logic := '0'; inclk0 : in std_logic := '0'; c0 : out std_logic); end component;</pre> | component pll is port (refclk : in std_logic rst : in std_logic outclk_0 : out std_logic); end component pll; |

component simple_counter port (clock : in std_logic; : in std_logic; reset counter_out : out unsigned(31 downto 0)); end component; component counter_bus_mux port (dataa : in unsigned(3 downto 0); datab : in unsigned(3 downto 0); sel : in std_logic; result : out unsigned (3 downto 0)); -- PLL Intel FPGA IP デザイン (Beryll Cyclone V GX end component; -- Base Board)の場合は、左の点線枠内を下記に - 変更してください。 begin pll_inst : pll port map (pll inst : pll port map (т rst => not RESET, I т areset => not RESET, Т refclk => OSC_CLK, inclk0 => OSC CLK, I. outclk 0 => pll c0 => pll c0 c0 1 Т I); т); simple_counter_inst : simple_counter port map (clock => pll_c0, => RESET, reset counter_out => cnt); counter bus mux inst: counter bus mux port map (=> cnt(24 downto 21), dataa => cnt(26 downto 23), datab => BUTTON, sel result => tmp); LED <= not tmp; end rtl;

- 4. File メニュー ➤ Save (または Ctrl + S) により my_first_fpga.vhd を保存します。
- 5. my_first_fpga.vhd のタブの端にある Close ボタンをクリックし、ファイルを閉じます。

以上ですべてのデザイン作成が完了しました。



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。

| | irst _/ Settings - my_first_fpga Category: | | | | |
|--|---|---|---|----------|---|
| Project Navigator | General | Files | | | |
| Files Counter_bus_mux.vhd Simple_counter.vhd Ell.gip | Files Literations I P Settings IP Catalog Search Locations Design Templates | Select the design files you the project. | want to include in the | project. | Click Add All |
| wHD my_first_fpga.vhd | Operating Settings and Conditions | | | | |
| | Temperature | File Name | Туре | Libran | Design Ent |
| | CG 連動している EDA Tool Settings Design Entry/Synthesis Simulation | counter_bus_mux.vhd simple_counter.vhd ▷ pll.qip my_first_fpga.vhd | VHDL File VHDL File IP Variation File (.qip) VHDL File | | <none> <none> <none> <none></none></none></none></none> |

余分なファイルや不足があった場合には、Project メニュー ➤ Add/Remove Files in Project により追加あ るいは削除を行ってください。

また、今回の VHDL デザインには、一部 VHDL-2008 に対応した記述があります。Assignments メニュー ➤ Settings ➤ Compiler Settings ➤ VHDL Input において、VHDL 2008 へ切り替えてください。

| Settings - my_first_fpga | |
|---|--|
| Category: | Device/Board |
| General | VHDL Input |
| Files Libraries P IP Settings | Options for directly compiling or simulating VHDL input files. (Click on the EDA Tool Settings category to enter options for VHDL files generated by other EDA tools.) |
| Design Templates Operating Settings and Conditions Compilation Process Settings EDA Tool Settings | © VHDL 19 <u>8</u> 7 © VHDL 19 <u>9</u> 3 |
| Compiler Settings VHDL Input | VHDL 2008 |

2-1-3. Analysis and Elaboration の実行

Processing メニュー ➤ Start ➤ Start Analysis and Elaboration をクリックし、コードのシンタックス・チェックや回路のデータベースを作成します。

| ments | Pro | cessing Tools Window Help | | | | |
|----------|-----|-----------------------------------|--------------|-------------------------------|---------|--------|
| [my_fi | 9 | Stop Processing | Ctrl+Shift+C | | | |
| <u>s</u> | | Start Compilation | Ctrl+L | | | |
| + QI | | Analyze Current File | | y_first_fpga | × | 40- |
| | | Start | ٠ | 📐 Start Hierarchy Elaboration | e. N | |
| | - | Update Memory Initialization File | | Start Analysis & Elaboration | n | |
| | 4 | Compilation Report | Ctrl+R | 🕺 Start Analysis & Synthesis | | Ctrl+K |

もしエラーが発生した場合は、VHDL コードに問題があります。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した内容を確認しましょう。 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もあります。ワーニングは必ず内容を 確認し、その内容を回避すべきか無視できるのかを仕様に応じて自己判断してください。

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。(次頁図を参照)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ので、その情報からエラーなどを回避してく ださい。



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 メニュー ➤ Save)し、再び Start Analysis and Elaboration を実行します。エラーやワーニングを回避できたら、ここでの作業は完了です。

以上で論理回路の設計は完了です。

2-2. 論理シミュレーションを実行する

2-1-2 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。)



ALTIMA

シミュレーションを行うには、FPGA に対して入力する信号と出力した信号をモニタする内容を 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 ルを入手した WEB ページからダウンロードしてください。その後、作業しているプロジェクトのディレクトリに保 存してください。

Note: ・ テストベンチの設計経験がないユーザは、以下をご活用ください。

<u>はじめてみよう!テストベンチ</u>

シミュレーション検証するには、インテル FPGA をサポートしている他メーカの言語シミュレータが必要です。 このチュートリアルでは、ModelSim* - Intel® FPGA Edition を使用して作業を進めます。

通常は ModelSim - Intel FPGA Edition を起動しオペレーションしますが、このチュートリアルでは "NativeLink 機能" により ModelSim - 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。

Notes: NativeLink とは、Quartus Prime のバックグラウンドで EDA ツールを操作・実行することを示します。 例えば、ModelSim - Intel FPGA Edition の GUI オペレーションを知らなくてもユーザは Quartus Prime 上の GUI 設定をするだけでシミュレーションが実行できるため、作業フローが簡素化され ます。また、IP を含んだデザインの論理シミュレーションをする場合、IP 用のシミュレーション・ライ ブラリのセットアップが必要になりますが、NativeLink でシミュレーションを実行するとそれらの環境 構築もスクリプトにより自動的にセットアップされるため、ユーザへのストレスが軽減されます。

詳細は下記ドキュメントをご参考ください。

[Using NativeLink Simulation (Intel Quartus Prime Standard Edition)]

Intel Quartus Prime Standard Edition Handbook Volume 3 Verification $\,\mathcal{LV}$

 ModelSim - Intel FPGA Edition を起動して GUI 操作によりシミュレーションを実行する場合は、下記 ドキュメントをご参考ください。

「ModelSim - Intel FPGA Edition - RTL シミュレーションの方法」

※ <u>ModelSim - Intel FPGA Edition の GUI が起動している場合は、File メニュー ➤ Quit で終了しておき</u> <u>ます。</u>

 Quartus Prime の Tools メニュー ➤ Options を選択し、Category 内の General ➤ EDA Tool Options をクリ ックします。ModelSim-Altera 欄に、ModelSim - Intel FPGA Edition の実行プログラム(modelsim.exe)が格 納されているディレクトリのパスを設定します。(インストールされているディレクトリが異なる場合は、使用す るパソコンに応じて変更してください。)

| ▲ General | EDA Tool Options | | |
|-------------------------------------|------------------------------|---|--|
| EDA Tool Options | Specify the directory that o | contains the tool executable for each third-par | tv EDA tool |
| Fonts Norders & Easters Sattings | EDA Tool Directo | bry Containing Tool Executable | 94 8 -19-21-02-02-02-02-02-02-02-02-02-02-02-02-02- |
| Processing | QuestaSim | | L |
| Tooltip Settings | ModelSim-Altera C:\inte | IFPGA\18.0\modelsim_ae\win32aloem | |



2. (Beryll Cyclone V GX Base Board $\neg - \# \mathcal{O} \mathcal{A}$)

Assignments メニュー ➤ Settings ➤ Files (Category 内)を選択し、PLL IP のシミュレーションを NativeLink で行うため、ここへ pll.sip ファイルが登録されていることを確認します。

| Settings - my_first_fpga | | | | |
|--|--|---|--|------------------|
| Category: | | | (| Device/Board |
| General | Files | | | |
| Files Libraries | Select the design files you project directory to the pr | ı want to include in oject. | the project. Click Add All to add all design | files in the |
| IP Settings IP Catalog Search Locations Design Templates | <u>F</u> ile name: | | | Add |
| Operating Settings and Conditions Voltage | File Name | Туре | Library Design Entry/Synthesis T | Add A <u>l</u> l |
| Temperature Compilation Process Settings Incremental Compilation | simple_counter.vhd counter_bus_mux.vh | VHDL File d VHDL File IP Variation File (| pll.sip ファイルが登録されてい には、ブラウズボタンにより登録 | ない場合 剥ます。 |
| EDA Tool Settings | pll.sip | Quartus Prime S | IP File <none></none> | Down |
| Design Entry/Synthesis | my_first_fpga.vhd | VHDL File | <none></none> | Properties |

3. Assignments メニュー ➤ Settings ➤ EDA Tool Settings ➤ Simulation をクリックします。各項目を設定してく ださい。

| EDA Simulation Settings > Simulation | |
|--------------------------------------|------------------------|
| Tools name | ModelSim-Altera |
| Format for output netlist | VHDL |
| NativeLink settings セクション | Compile test bench を選択 |

| itegory. | Device/Boar |
|---|---|
| General | Simulation |
| Files Libraries | Specify options for generating output files for use with other EDA tools. |
| IP Settings | Tool name: ModelSim-Altera |
| IP Catalog Search Locations Design Templates | Run gate-level simulation automatically after compilation |
| Operating Settings and Conditions | EDA Netlist Writer settings |
| Voltage Temperature | Eormat for output netlist: VHDL Time scale: 1 ps |
| Compilation Process Settings Incremental Compilation | Output directory: simulation/modelsim |
| EDA Tool Settings | Map illegal <u>H</u> DL characters Enable glitch filtering |
| Design Entry/Synthesis | Options for Power Estimation |
| Simulation | Concerto Value Change Dump A/CD) file script |
| Board-Level | |
| Compiler Settings VHDL Input | Design instance name: |
| Verilog HDL Input Default Parameters | More EDA Netlist Writer Settings, |
| TimeQuest Timing Analyzer | NativeLink settings |
| Assembler | O None |
| Design Assistant | |
| Signal Tap Logic Analyzer | <u>Compile test bench:</u> <u>Test Benches</u> |
| Logic Analyzer Interface | Use script to set up simulation: |



Test Benches ボタンをクリックします。Test Benches ウィンドウの New ボタンをクリックし、New Test Bench Settings ウィンドウにテストベンチ testbench.vhd の情報を入力します。

| ing test be | nch settings: | | | | <u>N</u> ew |
|-------------|------------------|-----------------|---------|------------------|--------------|
| Name | Top Level Module | Design Instance | Run For | est Bench File(s | <u>E</u> dit |
| | | | | | Delete |
| | | | | | - Zeven |
| | | | | | |
| | | | | | |

| New Test Bench Settings | | | | |
|--------------------------------|---|--|--|--|
| Test bench name | testbench | | | |
| Top level module in test bench | testbench | | | |
| Simulation period セクション | End simulation at: 500 と入力 / 単位: ms を選択 | | | |

| A New Test Bench Settings | × |
|--|---|
| Create new test bench settings. | |
| Test bench name: testbench | |
| Top level module in test bench: testbench | |
| Use test bench to perform VHDL timing simulation | |
| Design instance name in test bench: NA | |
| Simulation period | |
| Run simulation until all vector stimuli are used | |
| | |

Test bench and simulation files セクションの File name 右横にあるブラウズボタンをクリックします。Select File ウィンドウにおいて作業ディレクトリを開き、WEB ページから入手したテストベンチ・ファイル (testbench.vhd)を選択後 Open ボタンをクリックし、Add ボタンをクリックして登録します

| ile name: | | [| |
|---------------|---------|-------------|-------------|
| File Name | Library | HDL Version | Remove |
| testbench.vhd | | Default | Up |
| | | | Down |
| | | | Properties. |



すべての設定画面を OK ボタンで閉じます。

4. Tools メニュー ➤ Run Simulation Tool ➤ RTL Simulation を選択し、シミュレーションを実行します。この時、 ModelSim - Intel FPGA Edition の GUI が起動していないことを確認してください

| Тоо | ls Window Help | | |
|-----|------------------------------------|----|-----------------------|
| | Run Simulation Tool | N. | RTL Simulation |
| ~ | Launch Simulation Library Compiler | R. | Gate Level Simulation |

自動的に ModelSim - Intel FPGA Edition が起動し、シミュレーションが実行されます。画面の左下に、 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。

view signais .main pane.objects.interior.cs.body.tree run 500 ms Note: Cyclone 10 LP PLL locked to incoming clock Time: 90 ns Iteration: 3 Instance: /testbench/u1/pll_inst Now: 1,848,470 ns Delta: 5 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 ModelSim - 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から再度 Tools メニ ュー ➤ 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_pllc0、monia、 monib 信号を見てください。これらは、my_first_fpga エンティティの内部信号をモニタしたものです。
 - moni_pllc0 : PLL 回路の出カクロック
 - monia : カウンタ回路(32bit カウンタ)のレジスタ出力 [cnt(24 downto 21)]
 - monib : カウンタ回路(32bit カウンタ)のレジスタ出力 [cnt(26 downto 23)]

このチュートリアルで使用する開発キットに搭載されているクロックは 50MHz(20ns) です。デザインで は、50MHz を FPGA へ入力し、内部に構築した PLL のリファレンス・クロックとしています。また、PLL によ り 5MHz(200ns) のクロックを生成させ、カウンタ回路のクロックに用いています。



Wave ウィンドウにおいて、PLL の出力クロック(moni_pllc0)が正常に生成されていることを確認します。

| 8 • 🗃 🖬 🛸 🎒 } | (BB 22) | ○ - A :: < 2 :: 100 ps + 10 ps |) 🕅 💿 I 📰 🖄 |
|--|-----------------|----------------------------------|---------------|
| 3• • •€ • 📴 Search: | | | |
| | Msgs | \rightarrow \leftarrow | |
| /testbend /OSC_CLK /testbench/RESE1 /testbench/RESE1 | 1 1 | | |
| + | 0000 | 0000 | |
| /testbencl./moni_pllc0 | 2 | | |
| 🛨 🛶 /testbench/monia | 0000 | | |
| + | . 0000 | _0000 | |
| | | | |
| | | 200ns | |
| 🛤 🔍 Now | 50826794750 ps | 424600000 ps 424620000 ps | 1 1 1 1 1 1 1 |
| Cursor 1 | 42460 10000 ps | 4246010000 ps 20000 ps | |
| Cursor 2 | 4246030000 ps | 4246030000 ps | |
| Cursor 3 | 4246090000 ps | 200000 ps | |
| Cursor 4 | 4246290000 ps | | 4246290000 ps |

同様に、カウンタ回路(monia、monib)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を確認します。(ファンクション・シミュレーションは遅延を考慮しないため、クロック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. ModelSim Intel FPGA Edition 上の Simulate メニュー ➤ End Simulation を選択し、シミュレーションを終 了します。
- 7. File メニュー ➤ Quit を選択し、ModelSim Intel FPGA Edition を終了します。

以上で、作成したデザインの論理シミュレーションは完了です。

2-3. 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。

- 論理回路の入出カピンに、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 [ユーザ I/O ピンのアサイン]
- 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]

2-3-1. ユーザ 1/0 ピンのアサイン

ピンの割り当てを行うためには、Analysis & Elaboration の実行が必要です。このチュートリアルでは 2-1-3 章 ですでに実行していますのでスキップできます。未完了の場合には、以下のメニューから実行してください。

1. Processing $\checkmark = = - \rightarrow$ Start \succ Start Analysis & Elaboration

Analysis & Elaboration の実行後にエラーが発生した場合は、2-1-3 章を参考にエラーを回避してください。

デザイン上に存在するピンは、以下のとおりです。

| 入力ピン | OSC_CLK、RESET、BUTTON |
|-------|-----------------------------|
| 出ッカピン | LED[3]、LED[2]、LED[1]、LED[0] |

2. これらにピン番号をアサインするため、Pin Planner を起動します。

Assignments $\checkmark = = - \rightarrow$ Pin Planner





3. All Pins リストにデザイン上のピンが表示されていることを確認してください。各ピン名の横にある Location カラムに、下表のとおりピン番号を指定します。

| ピン・アサイン | 情報 | | |
|---------|-------------------------------|---|---|
| ピン名 | MAX 10 FPGA Evaluation Kit | <u>Cyclone 10 LP</u> <u>Evaluation Kit</u> | <u>Beryll Cyclone V GX</u> <u>Base Board</u> |
| BUTTON | 120 | D9 | V12 |
| LED[3] | 140 | J13 | J20 |
| LED[2] | 135 | J14 | H20 |
| LED[1] | 134 | K15 | G20 |
| LED[0] | 132 | L14 | H19 |
| OSC_CLK | 27 | E1 | U12 |
| RESET | 121 | E15 | AB24 |

Note ・ 各ピン番号が該当のボードのどのタクトスイッチや DIP スイッチ、LED なのかを、リファレンス・マ ニュアルやスケマティックで確認してください。

| × | Named: • • | 🐝 Edit: 🔀 🛷 | | ダブルクリックし、プルダウン・リスト からピン番号を選択 | |
|-----|---------------------------|-------------|----------|---|----------|
| 8 | Node Name | Direction | Location | /O Bank Current Strength I/O Standard | Reserved |
| 4 | BUTTON | Input | PIN D9 | | |
| | out LED[3] | Output | PIN_D9 | IOBANK_7 Column I/O DIFFIO_T15p | |
| | out LED[2] | Output | PIN_D11 | IOBANK_7 Column I/O DIFFIO_T23n | |
| | out LED[1] | Output | PIN_D12 | IOBANK_7 Column I/O DIFFIO_T23p | |
| | out LED[0] | Output | PIN_D14 | IOBANK_7 Column I/O DIFFIO_T24p | |
| | SC_CLK | Input | PIN_D15 | IOBANK 6 Row I/O | |
| | RESET | Input | PIN D16 | IOBANK 6 Row I/O | |
| | < <new node="">></new> | | PIN_E1 | IOBANK_1 Dedicated Clock CLK1, DIFFCLK_0n | |
| | | | PIN_E2 | IOBANK_1 Dedicated Clock CLK0, DIFFCLK_0p | |
| un. | | | PIN_E6 | IOBANK_8 Column I/O DIFFIO_T6p, DATA6 | |
| Pin | | | PIN_E7 | IOBANK_8 Column I/O DATA5 | 10 A |

4. 各ピン名の横にある I/O Standard カラムに、下表のとおり I/O 規格を指定します。

| I/O 規格情報 | | | |
|----------|-------------------------------|---------------------------------|-----------------------------------|
| ピン名 | MAX 10 FPGA Evaluation Kit | Cyclone 10 LP Evaluation Kit | Beryll Cyclone V GX Base Board |
| BUTTON | 3.3-V LVTTL | 3.3-V LVTTL | 1.5 V |
| LED[3] | 3.3-V LVTTL | 3.3-V LVTTL | 3.3-V LVTTL |
| LED[2] | 3.3-V LVTTL | 3.3-V LVTTL | 3.3-V LVTTL |
| LED[1] | 3.3-V LVTTL | 3.3-V LVTTL | 3.3-V LVTTL |
| LED[0] | 3.3-V LVTTL | 3.3-V LVTTL | 3.3-V LVTTL |
| OSC_CLK | 3.3-V LVTTL | 3.3-V LVTTL | 1.5 V |
| RESET | 3.3-V LVTTL | 3.3-V LVTTL | 3.3-V LVTTL |

関連情報: 下記ドキュメントをご参考ください。

<u>「Quartus Prime はじめてガイド - ピン・アサインの方法」</u> 「Quartus はじめてガイド - デバイス・オプションの設定方法」

2-3-2. 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の制 約を行います。

- 1. Tools メニュー ➤ Timing Analyzer を選択し、Timing Analyzer ウィンドウを表示させます。
- 2. File メニュー≻ 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、以下のコードを入力してください。

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



- 4. File メニュー ➤ Save As... により、SDC ファイルに名前を付け、保存します。
 - ・ 保存するディレクトリ: C:¥intelFPGA_prj¥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
- 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。



関連情報: 下記ドキュメントをご参考ください。

<u>「Quartus はじめてガイド - タイミング制約の方法」</u>

以上で、デザインに対する制約設定は完了です。



2-4. 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。)



1. デザインが完成したらコンパイルを実行します。

Processing メニュー≻ Start Compilation

コンパイルでは、論理合成や配置配線、タイミング解析、プログラミング・ファイルの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、本チュートリアル 2-1-3. Analysis and Elaboration の実行(21ペー

ジ)を参考にして、対処してください。

2. Full Compilation was successful のメッセージが確認できたら、コンパイル終了です。

| The car they respect Assignments rive | essing Tools Window Help | | Search altera.com |
|--|--|--|--|
| □ た 日 + □ 1 へ へ my_first_fp | ga 🔹 🖌 🎸 🚸 🔘 I | | |
| 💠 🔁 🗟 🦃 🐥 🥖 班 🕥 🐁 🏷 🐓 | 🐅 💠 ኛ 💐 💐 🗇 🛥 🗟 🔖 👗 | | |
| Project Navigator | २ 🗊 🗗 🗴 😓 Compilation Rep | ort - my_first_fpga 🛛 🛛 🔯 | |
| Entity:Instance | Table of Contents | Flow Summary | |
| Cyclone 10 LP: 10CL025YU256I7G | Flow Summary | < <filter>></filter> | |
| Image: my_first_fpga | Flow Settings | Flow Status | Successful - Fri Jun 22 15:14:46 2018 |
| | Flow Non-Default Global Setti | n Quartus Prime Version | 18.0.0 Build 614 04/24/2018 SJ Standard Edition |
| | Flow Elansed Time | Revision Name | my_first_fpga |
| 21 | Fld 💟 Quartus Prime | | my_first_fpga |
| , | Full Compilation | on was successful (10 warnings) | Cyclone 10 LP |
| Tasks Compilation | | | Tional Club Club Club Club Club Club Club Clu |
| Task | | ОК | 29 / 24 624 (< 1.04.) |
| 🥜 🔺 🕨 Compile Design | 1 III.) 🚬 "I | | |
| | | | |
| and a second | | | |
| < | | | |
| × All 🔞 🔊 🚺 🔊 👻 <<-Filter>> | 💏 Fir | nd | |
| | 😝 Fir | nd 😥 Find Next | |
| M All O A A Type ID Message Command: quartus ed | aread settinos files=offwri | nd) 💏 Find Next | rst foga -c mv first foga |
| ★ All ② ▲ ▲ ← Type ID Message Command: quartus_ed 18236 Number of processor | aread_settings_files=offwrii s has not been specified which may | nd 🏀 Find Next te_settings_files=off my_fi y cause overloading on shar | rst_fpga -c my_first_fpga ed machines. Set the global assignment NUM_PARALL |
| ★ All ② ▲ ▲ ★ Type ID Message O Command: quartus_ed 18236 Number of processor 10905 Generated the EDA f | <pre>aread_settings_files=offwrin s has not been specified which may unctional simulation netlist becan</pre> | nd Ø Find Next te_settings_files=off my_fi y cause overloading on shar use it is the only support | rst_fpga -c my_first_fpga ed machines. Set the global assignment NUM_PARALL d netlist type for this device. |
| <pre>x all ② ④ ▲</pre> | <pre>laread_settings_files=offwrin s has not been specified which may unctional simulation netlist becar irst_fpga.vho in folder "c:/intell etlist writer was successful. 0 er</pre> | <pre>nd</pre> | rst_fpga -c my_first_fpga ed machines. Set the global assignment NUM_PARALL d netlist type for this device. lation/modelsim/" for EDA simulation tool |
| All O A A A A A A A A A A A A A A A A A | <pre>laread_settings_files=offwrin s has not been specified which may unctional simulation netlist becar irst_fpga.vho in folder "c:/intell etlist writer was successful. 0 erro Compilation was successful. 0 error</pre> | <pre>hd Of Find Next te_settings_files=off my_fi y cause overloading on shar use it is the only supporte FPGA_prj/my_first_fpga/simu rors, 2 warnings ors, 10 warnings</pre> | rst_fpga -c my_first_fpga ed machines. Set the global assignment NUM_PARALL d netlist type for this device. lation/modelsim/" for EDA simulation tool |

コンパイルが終了すると、コンパイル・レポート(Compilation Report)にコンパイル結果が表示されます。

コンパイルでは、Analysis & Synthesis、Fitter、Assembler、Timing Analyzer の4工程が基本的に実行され、工程ごとにレポートが生成されます。ユーザは、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、下記ドキュメントをご覧ください。

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

関連情報: 下記ドキュメントをご参考ください。

「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」

2-5. タイミングを検証する

コンパイルの配置配線結果が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検証し、ユーザの要求値を満足(メット)しているかどうか を Timing Analyzer レポート(コンパイル・レポート内)で確認することができます。

実際の設計においては、タイミング がユーザの要求値を満足していない と実機でも動作しないため、非常に重 要な作業工程です。まずコンパイル・ レポートで検証結果のサマリを確認し、 メットできていない箇所があった場合 に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。

| | Slow T200mV 1 | ODC Model | | | | | | | | | |
|---|-------------------|---------------|------------|-------------------|--------------------------|--------|------------------------------|----------------|----------------|--------------|-----------------------------|
| Advanced in Timing | Command Info | Summary o | Patra | | | | | | | | |
| SDC File List | Slock | - | From Node | | To N | ode | | | lainth(C | ork - | |
| * 🖿 Summary (Setup) | t line and | simple counts | amela com | tar instreet 1 | simila countersimila o | - | insticuti26 | instituted on | monoranti auto | menarated | initiation and installed |
| Setup pl_restjatpl_componentiaux | 2 195.852 | same rante | sande ente | der textionent | smele courder smele a | meda | institutized at | instalted on | and a state | erreraintial | United in the light |
| Slow 1300mV 100C Model 🔹 | 3 196 203 | smile courte | sinde cou | ter insticutio | single courtersingle o | mante | le lestent251 al | instalial co | macoentiauto | repeateding | icition al installati com |
| · · · · · · · · · · · · · · · · · · · | 4 195.707 | emple courts | emple cour | ter insticut[1] | single countersingle of | ounte | insticut[24] of | instialtal co | macrentiauto | peneratedipl | (cik)0) pil instististi com |
| Report Set Opprating Conditions | 5 195.712 | ample courte | ample cour | tar insticut(3) | single countersingle of | ounter | instication | instituted co | moorentiauto | generatediol | Icikio] ol instalatol com |
| Tanion 🗐 🖉 🖛 = | 6 196.723 | smple counte | smple_cour | ter insticut?) | simple courtersimple o | ounte | insticut251 pl | instiated co | macrentjasto | pereratedial | idildi pil instalipil con |
| 🖌 👻 Open Project | 4 | | | | | | - | | | | |
| The Netlist Setup | Path #1: Setun a | iekis 196323 | | | | | Park at School | rik in 198,572 | , | | |
| Create Timing Netlini | Path Summary | Shatistica | Data Path | Wanform | Extra Either Information | | Talk Summary | Sheltones | Data Dath | Woweform | Patra Titler Information |
| Read SOC File | Date Arrivel Date | distantianes. | | Louis Contraction | | 1 | | | | | |
| Beat Desirp | Contract to | | 1 | 2 | | 14 | | | | | |
| Set Operating Conditions | Total | na | R | Type San | out Location | -0 | | | | | |
| Peporta | 1 0.000 | 0.000 | | | | | | | | | |
| - P stack | 2 4 0.315 | 0.318 | | | | | Laurch Clock ^{1,80} | 10 | | | |
| Report Setup Summary | 1 0.00 | 0 0.000 | | | - | | | 1 | | 230.1 100 | |
| Report Hold Summary | 4 | 0 | | 1 | PER PT | + | 284.00 #E10010+0 | | | | |
| Report Recovery Summary | Data Required P | ath | | | | | Latch Clock | _ | | 1 | ALCE. |
| Report Minimum Pulse with Sur | Total | 100 | BF. | Tuge Fat | out Location | | 120203-006 | | _ | - | |
| The second second second second | 1 200.000 | 200.00 | 1 | dise | | -01 | Data Perival | X | | | |
| Report Maa seave surranary | | 0.048 | | | | | anosana a | adam. | | | |
| Report Net Delay Summary | 12 4 2081245 | | | | | | C108-19146 | | | | |
| Report Net Delay Summary | 2 4 200.248 | 000 0.000 | | | | | | | | | |
| Report Not Delay Summary Delay Summary Delay Export Finae Summary | 2 4 200,248 | 000 0 000 | | | DN F1 | | taka dalay | 1.10 | | | |

今回は、タイミング制約内容を満足

する仕様になっていることと、Quartus Prime の基本的なオペレーションを体験するためのチュートリアルなので、 タイミング検証は割愛します。 下記に関連情報をご案内しますので、次のステップとして習得してください。

関連情報: 下記ドキュメントをご参考ください。

「Quartus Prime はじめてガイド - TimeQuest によるタイミング解析の方法」

タイミング検証のトレーニングも開催されています。

<u>インテル FPGA テクニカル・トレーニング「Quartus Prime パーフェクト・コース II : タイミング解析」</u>

2-6. プログラミング

FPGA ヘデータを書き込みます。(この作業は、開発フローの [8] にあたります。)

| 1準備 | | | | 開 | 発 | | | | 10 |
|----------------------------|-----------|-----------|------------------|-----------------|----------------|------------------------|-----------|--------|----|
| ・ ダウンロード ・ ダウンロード | 2機能仕様の検討・ | 3 論理回路の設計 | 4 論理シミュレーション | 5 制約の設定 ● | 6 コンパイル | 7 タイミング検証 | 8 プログラミング | 9 実機検証 | 量産 |
| | | Model | Sim* - Intel' Fl | PGA Edition | Inte Design | l' Quartu: Software | s' Prime | | |

デザインをコンパイルしてタイミングを検証したら、開発ボード上の FPGA を実際に動作させる準備が整いました。作成したファイルをダウンロードします。

Note: · Cyclone 10 LP Evaluation Kit をご利用の場合は、事前に DIP スイッチ [SW1] を下図のとおりに変更してから作業を行ってください。

| | | | WE | |
|---|---|---|--------|--|
| 1 | 2 | 3 | 山 4 | |

プログラミング・データは、コンパイル実行時に作成され、FPGA の場合 .sof ファイルが生成されます。

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 用のコンフィギュレーション・デバイス(FPGA が起動するための 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。

コンフィギュレーション・デバイスへ書き込んで実機検証を行っても良いのですが、FPGA への書き込み時間と 比べて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 の実機検証を行い、修正が無いことを確認した上でコンフィギュレーションにプログラムすることを推奨し ています。

<u>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ギ</u> <u>ュレーション・デバイスへ書き込みを行います。</u> 2-6-1. sof ファイルを書き込む

- 1. ダウンロード・ケーブルとパソコンを接続します。
 - Note: ・ 開発ボードの JTAG ヘッダの位置は、各開発ボードのユーザ・ガイドやスケマティックをご確認く ださい。
 - MAX 10 FPGA Evaluation Kit の場合

インテル FPGA ダウンロード・ケーブル II と USB ケーブルを接続し、インテル FPGA ダウンロード・ ケーブル II を開発ボードの JTAG ヘッダに接続します。その後 USB ケーブルのもう一方をプログラ ミングに使用するパソコンの USB ポートに接続します。

- Cyclone 10 LP Evaluation Kit の場合

ここでの作業はありません。次のステップへ進んでください。

(Cyclone 10 LP Evaluation Kit は、ダウンロード・ケーブルと電源供給ケーブルが同じ USB ケーブルで 賄われています。)

- Beryll Cyclone V GX Base Board の場合

付属の USB ケーブルで開発ボードとパソコンを接続します。開発ボード側のコネクタ位置は、本開発 ボードの Getting Started をご覧ください。

- 開発ボードと電源ケーブルを接続します。(開発ボードによっては、電源ケーブルを接続すると電源が供給 されます。)電源スイッチがある開発ボードの場合は、スイッチを入れて電源を供給します。
- 3. Tools メニュー ➤ Programmer により、Programmer ウィンドウを起動します。

| File | Device | Checksum | Usercode | Program/ Configure | Verify | Blank Check |
|--------------------------------|--------------|----------|----------|-----------------------|--------|----------------|
| output_files/my_first_fpga.sof | 10CL025YU256 | 00138EF2 | 00138EF2 | V | | |
| t | | | | | | |
| | | | | | | |
| | | | | | | |
| <u> </u> | | | | | | |
| | | | | | | |
| | | | | | | |
| | ٦ | | | | | |
| | Ī | | | | | |

4. Hardware Setup ボタン(Programmer 画面の左上)をクリックします。



5. Hardware Settings タブの Currently selected hardware 項のプルダウン・リストから使用するハードウェアを 選択し、Close ボタンをクリックします。

| elect a programming hardware setup to use when programming devices. This programming ardware setup applies only to the current programmer window. urrently selected hardware: USB-BlasterII [USB-1] Available hardware items | elect a programmin ardware setup appli | g hardware setup | to use wher | n programming de | vices This programming |
|---|---|--------------------|----------------|------------------|------------------------|
| ardware setup applies only to the current programmer window. urrently selected hardware: USB-BlasterII [USB-1] Available hardware items | ardware setup appli | es only to the sur | | | vices, mis programming |
| urrently selected hardware: USB-Blasterii [USB-1] Available hardware items | SA 9656 | es only to the cur | rent progran | nmer window. | |
| Available hardware items | urrently selected ha | rdware: USB-B | lasterii [USB- | -1] | |
| | Available hardware | items | | | |
| | 1 | | Local | USB-1 | 6 |

| Board | Currently selected hardware に検出される名前 |
|--------------------------------|--------------------------------------|
| MAX 10 FPGA Evaluation Kit | 使用するダウンロード・ケーブル名 |
| Cyclone 10 LP Evaluation Kit | Cyclone 10 LP Evaluation Kit |
| Beryll Cyclone V GX Base Board | USB-Blaster |

6. Mode(Programmer 画面の右上)のプルダウン・リストより、デバイスへの書き込みを実行する手法を選択します。

このチュートリアルでは、JTAG を選択します。



7. Programmer 画面上部の File 欄には、セットしているプロジェクト用の my_fast_fpga.sof ファイルがすで に登録されています。また Device 欄には、開発している FPGA の型番が表示されています。





もし別の .sof ファイルが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。

その後 Add File ボタンをクリックし、作業ディレクトリ内の output_files フォルダ内に生成されている my_first_fpga.sof 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。

| Look in: | C:\intelFPGA_pa\output_files 🔻 🗿 🗿 | | |
|------------------------------------|--|-------|-----------|
| 鷆 Му | Name | Size | Туре |
| | my_first_fpga.sof | 687 K | B sof Fi |
| | | | |
| 4 <u>m</u> • | ۲ III) | | , |
| ∢ <u>m</u> ► File <u>n</u> ame: | ۲۰۰۰ ۲۰۰۰ ۲۰۰۰ ۲۰۰۰ ۲۰۰۰ ۲۰۰۰ ۲۰۰۰ ۲۰۰ | |) Open |

8. my_first_fpga.sof ファイル行の右側にあるプログラミング・オプションにおいて、実行する項目にチェックを 入れます。

このチュートリアルでは、FPGA に .sof データを転送するので、Program/Configure オプションにチェックを入れてください。

| ▶ [™] Start | File | Device | Checksum | Usercode | Program/ Configure | Verify | Blank- Check | Examine | Security Bit | Erase | ISP CLAMP |
|----------------------|---|--------------|----------|----------|-----------------------|--------|-----------------|---------|-----------------|-------|--------------|
| Matto Detect | outp <mark>ut_</mark> files/my_first_fpga.sof | 10CL025YU256 | 00138EF2 | 00138EF2 | V | | | | | | |
| X Delete | | | | | | | | | | | |

9. Start ボタンをクリックし、データの書き込みをスタートさせます。

データのダウンロードが開始されます。Progress バーが 100% になり、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書き込み完了です。

| M Start | File | Device | Checksum | Usercode | Program/ Configure | Verify | Blank- Check | Exan |
|-------------|--------------------------------|--------------|----------|----------|-----------------------|--------|-----------------|------|
| No Stop | output_files/my_first_fpga.sof | 10CL025YU256 | 00138EF2 | 00138EF2 | V | | | E |
| Auto Detect | *m | | | | | | | , |

続いて、2-7. 実機検証 を行い FPGA の動作を確認します。



2-6-2. コンフィギュレーション・デバイスに書き込む

この作業は、先に次項 2-7. 実機検証を行い、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。

2-6-2-1. 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ギュレーション・デバイス用のプログラミング・ファイルを sof ファイルから生成します。

| Board | Configuration Device | File Format |
|--------------------------------|----------------------|-------------|
| MAX 10 FPGA Evaluation Kit | MAX 10 内蔵 CFM | pof |
| Cyclone 10 LP Evaluation Kit | EPCQ64 | jic |
| Beryll Cyclone V GX Base Board | EPCS128 | jic |

なお、MAX 10 用の pof ファイルは、コンパイル実行時に同時生成されていますので、MAX 10 FPGA Evaluation Kit ユーザは <u>2-6-2-2. プログラミングする</u> へ進んでください。

- 1. File メニュー ➤ Convert Programming Files を起動します。
- 2. Output programming file 枠内を設定します。

| le <u>T</u> ools <u>W</u> indow | | Search altera.com |
|---------------------------------|--|----------------------------|
| pecify the input files to | convert and the type of programming file to generate | |
| ou can also import inpu | t file information from other files and save the conversion setup in | formation created here for |
| iture use. | | |
| Conversion setup files | | |
| · · | 3 | |
| Open (| Conversion Setup Data Save | Conversion Setup |
| Output programming fi | | |
| output programming n | | |
| Programming file type: | JTAG Indirect Configuration File (.jic) | |
| Options/Boot info | Configuration device: EPCQ64 Mode: | Active Serial 🔹 |
| File <u>n</u> ame: | 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_firs | t_fpga.core.rbf) |
| | | |

| 設定名 | | 内容 |
|--------------------|-----|--|
| Programming file t | уре | JTAG Indirect Configuration File (.jic) |
| Configuration devi | ce | Cyclone 10 LP Evaluation Kit の場合 : EPCQ64 |
| | | Beryll Cyclone V GX Base Board の場合 : EPCS128 |
| Mode | | Active Serial |
| File name | | output_files/my_first_fpga.jic |



3. Input files to convert 枠内を設定します。

ALTIMA

(1) 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。

| File/Data area | Properties | Start Address | Add Hex Data |
|----------------|------------|---------------|--------------|
| Flash Loader | | | Add Sof Pag |
| SOF Data | Page_0 | <auto></auto> | |

- 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・ファミリとデバイス名を選択し、OK ボタンをクリックします。
 - ▼ Cyclone 10 LP Evaluation Kit の場合

| Device family | Device name | | |
|-----------------|-------------|----------|-------------|
| APEX20K | 10CL006V | | New |
| 🗌 Arria 10 | 10CL0062 | 1 | 2 |
| Arria GX | 10CL010Y | | Import |
| 🗌 Arria II GX | 10CL0102 | <u>:</u> | Export |
| 🔲 Arria II GZ | E 10CL016Y | (| C |
| 🗌 Arria V | 10CL0162 | 1 | Edit |
| 🗌 Arria V GZ | M 10CL025Y | < | Persoue |
| Cyclone | 10CL0252 | | |
| 🗹 Cyclone 10 LP | 10CL040Y | ł | Uncheck All |
| 🗌 Cyclone II | 10CL0402 | <u>.</u> | |
| Cyclone III | 10CL055Y | | |
| Cyclone III LS | 10CL0552 | | |
| Cyclone IV E | 10CL080Y | · | |
| Cyclone IV GX | 10CL0802 | 1 | |
| Cyclone V | 10CL120Y | | |
| 🗍 11-147-1-11 | | | |







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。

| File/Data area | Properties | Start Address | Add Hex Data |
|----------------|------------|---------------|--------------|
| Flash Loader | | | Add Sof Page |
| 10CL025Y | | | |

④ 変換元の my_first_fpga.sof ファイルを指定します。

| .ook in: | C:\int | elFPGA_prj\my_first_fpg | a\output_files | - 0 0 | 0 | • |
|----------|--------|-------------------------|----------------|-------|-------|------------|
| 鷆 My Col | mputer | Name | * | | Size | Туре |
| | | my_first_fpga.sof | | | 687 K | B sof File |
| | | | | | | |

4. Generate ボタンをクリックして JIC ファイルを生成させます。



5. Close ボタンをクリックし、Convert Programming File ダイアログボックスを閉じます。

以上で、コンフィギュレーション・デバイス用のプログラミング・ファイルが完成しました。

2-6-2-2. プログラミングする

- 1. ダウンロード・ケーブルとパソコンを接続します。
 - Note: ・ 開発ボードの JTAG ヘッダの位置は、各開発ボードのユーザ・ガイドやスケマティックをご確認く ださい。
 - MAX 10 FPGA Evaluation Kit の場合



インテル FPGA ダウンロード・ケーブル II と USB ケーブルを接続し、インテル FPGA ダウンロード・ ケーブル II を開発ボードの JTAG ヘッダに接続します。その後 USB ケーブルのもうー方をプログラ ミングに使用するパソコンの USB ポートに接続します。

- Cyclone 10 LP Evaluation Kit の場合

ここでの作業はありません。次のステップへ進んでください。

(Cyclone 10 LP Evaluation Kit は、ダウンロード・ケーブルと電源供給ケーブルが同じ USB ケーブルで 賄われています。)

- Beryll Cyclone V GX Base Board の場合

付属の USB ケーブルで開発ボードとパソコンを接続します。開発ボード側のコネクタ位置は、本開発 ボードの Getting Started をご覧ください。

- 開発ボードと電源ケーブルを接続します。(開発ボードによっては、電源ケーブルを接続すると電源が供給 されます。)電源スイッチがある開発ボードの場合は、スイッチを入れて電源を供給します。
- 3. Tools メニュー ➤ Programmer により、Programmer ウィンドウを起動します。
- 4. Hardware Setup ボタン (Programmer 画面の左上)をクリックします。
- 5. Hardware Settings タブの Currently selected hardware 項のプルダウン・リストから使用するハードウェアを 選択し、Close ボタンをクリックします。

| ielect a programmi | ng hardwa | re setup to use whe | en programming d | evices. This programming |
|---|--------------|----------------------|---------------------|--------------------------|
| and the second | the make the | the survey of survey | | |
| ardware setup app | ues only to | o the current progra | mmer window. | |
| urrently selected h | ardware: | USB-Blasterii [USI | 3- <mark>1</mark>] | 1 |
| | | 20 | | |
| Available hardware | e items | | | |
| Available hardware | e items | Server | Port | Add Hardware |

| Board | Currently selected hardware に検出される名前 |
|--------------------------------|--------------------------------------|
| MAX 10 FPGA Evaluation Kit | 使用するダウンロード・ケーブル名 |
| Cyclone 10 LP Evaluation Kit | Cyclone 10 LP Evaluation Kit |
| Beryll Cyclone V GX Base Board | USB-Blaster |

Mode(Programmer 画面の右上)のプルダウン・リストより、デバイスへの書き込みを実行する手法を選択します。



このチュートリアルでは、JTAG を選択します。



- 7. Programmer 画面上部の 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。
- 8. Add File ボタンをクリックし、作業ディレクトリ内の output_files フォルダ内に生成されている my_first_fpga.pof あるいは my_first_fpga.jic 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。

| ► Start | File | Device | Checksum | Usercode | Program/ Configure | Verify |
|---------------------|---|--------------------|----------------------|-------------|-----------------------|--------|
| Stop | <none> output files/my first fpgajic</none> | 10CL025Y EPCQ64 | 00000000 74B3D0B8 | 0017EA71 | | |
| X Delete | | | | | 91253 | |
| Add File | EPCQ64 | 使用れてい | する開発ボー いる FPGA の雪 | ドに搭載さ 型番 | | |
| Add Device | | | | | | |
| Up [™] | Board | | File 欄 | | Devid | æ 欄 |
| L ^M Down | MAX 10 FPGA Evaluation Kit | output_f | iles/my_first_ | fpga.pof | 10M08 | SAE144 |
| | Cyclone 10 LP Evaluation Kit | | | | 10CL | 025Y |
| | | output_ | files/my_first_ | fpga.jic | EPC | Q64 |
| | Beryll Cyclone V GX Base Board | | | | 5CGXFC | 4C6F27 |
| | | output_ | files/my_first_ | fpga.jic | EPC | 5128 |

9. my_first_fpga.pof あるいは my_first_fpga.jic ファイル行の右側にあるプログラミング・オプションにおいて、 実行する項目にチェックを入れます。

ここでは コンフィギュレーション・デバイスにプログラミングするので、Program/Configure オプションにチェックを入れてください。

• MAX 10 FPGA Evaluation Kit の場合は、pof ファイルの下にある CFM0 側にチェックをしてください。

| No. Start | File | Device | Checksum | Usercode | Program/ Configure | Verify |
|--------------|---|-------------|----------|----------|-----------------------|--------|
| Matto Detect | output_files/my_first_fpga.pof CFM0 UFM | 10M085AE144 | 02667BF8 | 0008A690 | | |



Cyclone 10 LP Evaluation Kit / Beryll Cyclone V GX Base Board の場合は、jic 側にチェックしてください。自動的に上側の枠にもチェックが入力され、File 欄には "Factory default enhanced SFL image" が表示されます。

| ▶ [™] Start | File | Device | Checksum | Usercode | Program/ Configure | Verify |
|----------------------|------------------------------------|----------|----------|----------|-----------------------|--------|
| who Stop | Factory default enhanced SFL image | 10CL025Y | 0017EA71 | 0017EA71 | V | |
| Auto Detect | output_files/my_first_fpga.jic | EPCQ64 | 74B3D0B8 | | | |

10. Start ボタンをクリックし、データの書き込みをスタートさせます。

データのダウンロードが開始されます。Progress バーが 100% になり、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書き込み完了です。

関連情報: 下記ドキュメントおよび WEB コンテンツをご参考ください。 <u>
「Quartus Prime はじめてガイド - デバイス・プログラミングの方法」</u>

FPGA 経由で EPCQ デバイスヘプログラミング(JIC プログラミング)



2-7. 実機検証

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



チュートリアルでは、L チカ(LED チカチカ)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。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、あるいは DIP スイッチを切り替えると、点灯する速さが変わります。



<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している間、LED の点滅および点滅する 速さが変わり(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している間、PLL 回路とカウンタ回路がリセットされ、LED が全消灯します。

L チカは成功しましたか?

なお、各開発ボードのスイッチや LED の位置は、このとおりです。

▼ MAX 10 FPGA Evaluation Kit の場合

▼ Cyclone 10 LP Evaluation Kit の場合

▼Beryll Cyclone V GX Base Board の場合

以上で、FPGA 開発における Quartus Prime の基本的な操作は終了です。

3. <u>ステップアップ</u>

FPGA や CPLD を勉強するために便利な WEB サイトをご紹介します。

| サイト名 | 概要 |
|----------------------|---|
| インテル PSG ホームページ | メーカのホームページです。 |
| | <u>英語 日本語</u> |
| FPGA 入門 | FPGA について分かりやすく紹介しています。 |
| (インテル PSG HP より) | |
| <u>CPLD 入門</u> | CPLD について分かりやすく紹介しています。 |
| (インテル PSG HP より) | |
| <u>はじめての CPLD 設計</u> | CPLD 設計に役立つデザイン例 |
| (インテル PSG HP より) | |
| <u>サポート情報</u> | 開発ソフトウェアやトランシーバ、外部メモリ・インタフェースなど、カテゴリ |
| (インテル PSG HP より) | 別に情報がまとめられたページです。 |
| オンライン資料 | デバイスやツールのデータシートやユーザ・ガイドなどが閲覧できます。 |
| (インテル PSG HP より) | PDF としてダウンロードすることも可能です。 |
| ダウンロード・センター | インテル FPGA 開発に必要なソフトウェア各種がダウンロードできます。 |
| (インテル PSG HP より) | |
| <u>アルテラ フォーラム</u> | 他の FPGA ユーザに質問をしたり、問題を解決したり、アイデアを共有し たりできます。 |
| アルテラ Wiki | 組み込み設計者がデザイン・ガイドやツール、デザイン例等を共有できる |
| | コミュニティです。 |
| Design Store | インテル FPGA を使い始めるのに役立つサンプル・デザインの検索ペー |
| (インテル PSG HP より) | ジです。 |
| | ★関連情報★ |
| | 【TIPS】 サンブル・デザインを無料で手に入れよう |
| インストラクター・トレーニング | インテル 公式 FPGA テクニカル・トレーニングです。講師がレクチャーす |
| (インテル PSG HP より) | るハンズオン・トレーニングです。[有料] |
| オンライン・トレーニング | インテル公式オンライン・トレーニングです。インテル FPGA 製品に関す |
| (インテル PSG HP より) | る各種トレーニングをいつでもどこでも受講できます。[無料] |
| | |
| | (TIPS) インテル(旧アルテラ) FPGA テクニカル・トレーニングを無料で 受講しよう |
| MACNICA フォーラム | マクニカ在籍のエンジニアに質問したり、メンバー同士のアイデアや意見 |
| | 交換ができます。投稿および閲覧の際は、カテゴリより "Intel FPGA(旧 |
| | ALTERA)"を選択してご利用ください。 |
| | ※ はじめにお読みください |

この情報は下記 WEB コンテンツでも公開しています。

ほんとのほんとの導入編 その 5. FPGA 役立つコンテンツの紹介

改版履歴

| Revision | 年月 | 概要 |
|----------|------------|----|
| 1 | 2018 年 7 月 | 初版 |

免責およびご利用上の注意

弊社より資料を入手されましたお客様におかれましては、下記の使用上の注意を一読いただいた上でご使用ください。

- 1. 本資料は非売品です。許可無く転売することや無断複製することを禁じます。
- 2. 本資料は予告なく変更することがあります。
- 本資料の作成には万全を期していますが、万一ご不明な点や誤り、記載漏れなどお気づきの点がありましたら、本資料を入手されました下記代理店までご一報いただければ幸いです。
 株式会社マクニカ アルティマ カンパニー https://www.alt.macnica.co.jp/ 技術情報サイト アルティマ技術データベース https://www.alt.macnica.co.jp/
- 4. 本資料で取り扱っている回路、技術、プログラムに関して運用した結果の影響については、責任を負いかねますのであらかじめご了承ください。
- 5. 本資料は製品を利用する際の補助的な資料です。製品をご使用になる際は、各メーカ発行の英語版の資料もあわせてご利用ください。