

Efinity紹介

Efinix社FPGA用開発環境

※こちらの資料では、全46ページ中29ページまでの公開となります。続きの閲覧には、ダウンロードが必要となります。



本書の目的

- ✓ 対象者
 - FPGA開発フローの概略を知りたい方
 - これからEfinix[®] FPGA開発を始める方、検討している方
 - 他社製品からEfinix[®] FPGAへの乗り換えを検討している方

- ✓ 受講のゴール
 - Efinityの導入方法を知る
 - 入手方法、メンテナンス
 - FPGA開発フローを知る
 - Efinityに実際に触れる
 - 基本操作を学ぶ



ユーザー登録

ドキュメントやツールのダウンロードなどサポートページの利用にはユーザー登録が必要になります。

ユーザー登録は以下のページにアクセスし必要情報を入力します

➤ [Efinix Support \(efinixinc.com\)](https://www.efinixinc.com/support)

ユーザー登録が済み、ボード発注手続きが済みましたら、ツールダウンロードの権利を有したことになります。

ツールダウンロード可能にする手続きは以下のいずれかの方法で可能です。

- ✓ 弊社に（発注手続きが済んだ旨）ご連絡いただく
- ✓ ボード到着後、以下のページにて登録手続きを行う

➤ <https://www.efinixinc.com/support/profile-devkit.php>



Efinity入手方法

- ✓ Efinixウェブサイト
 - <https://www.efinixinc.com/support/index.php>
 - 上記サポートページにてログイン
 - 未登録の場合'Register'をクリック→必要項目を入力し登録を行います。
- ✓ ツールダウンロード権利の取得方法
 - Efinityをダウンロードするには以下の方法があります
 - 有償（新規\$150、~~更新\$100~~）で購入いただく
 - 評価ボードを購入いただく（紐づいてダウンロード権提供）
 - Efinityはライセンス管理をしていない（サポート期限を過ぎても利用可能）。



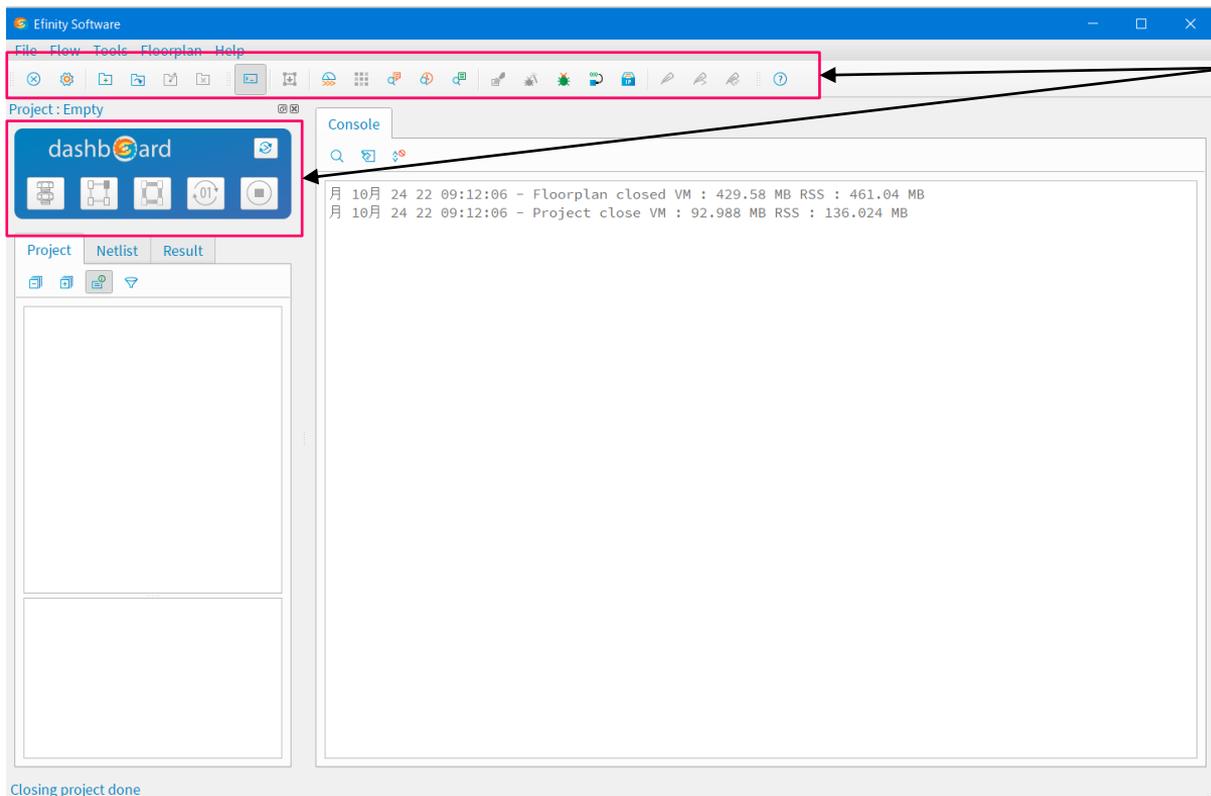
Efinityとは

- EfinityはEfinix社のFPGAを開発するためのソフトウェア（ツール）
- 論理回路の設計～デバイスへの書き込みまでをカバー（※）
※シミュレーションはModelSimなどのツールが必要
- 本書では
 - Efinityの機能概略説明を行ない「まずは使ってみる」を実践します
 - ✓ 今まで使っていたツールと違いわかりづらい
 - ✓ ユーザーガイド（以下、UG）が英語で読み込みが大変
 - ✓ UGの説明が詳細で何から手をつければ良いかわかりづらい

を解決することを目指しています



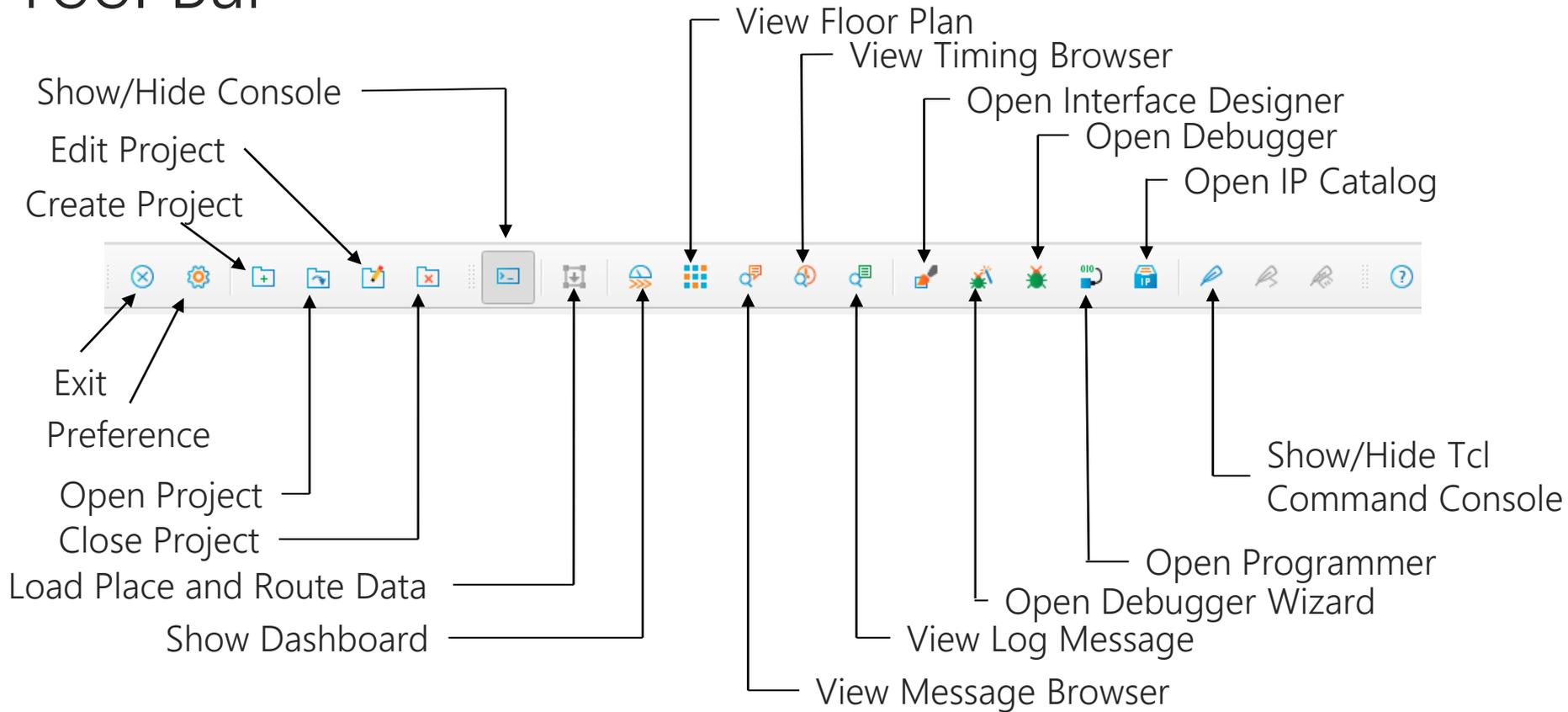
Efinity起動画面



◆ GUIベースのツール

ツールバーや
Dashboardで多くの
機能/フローを実行可能

Tool Bar

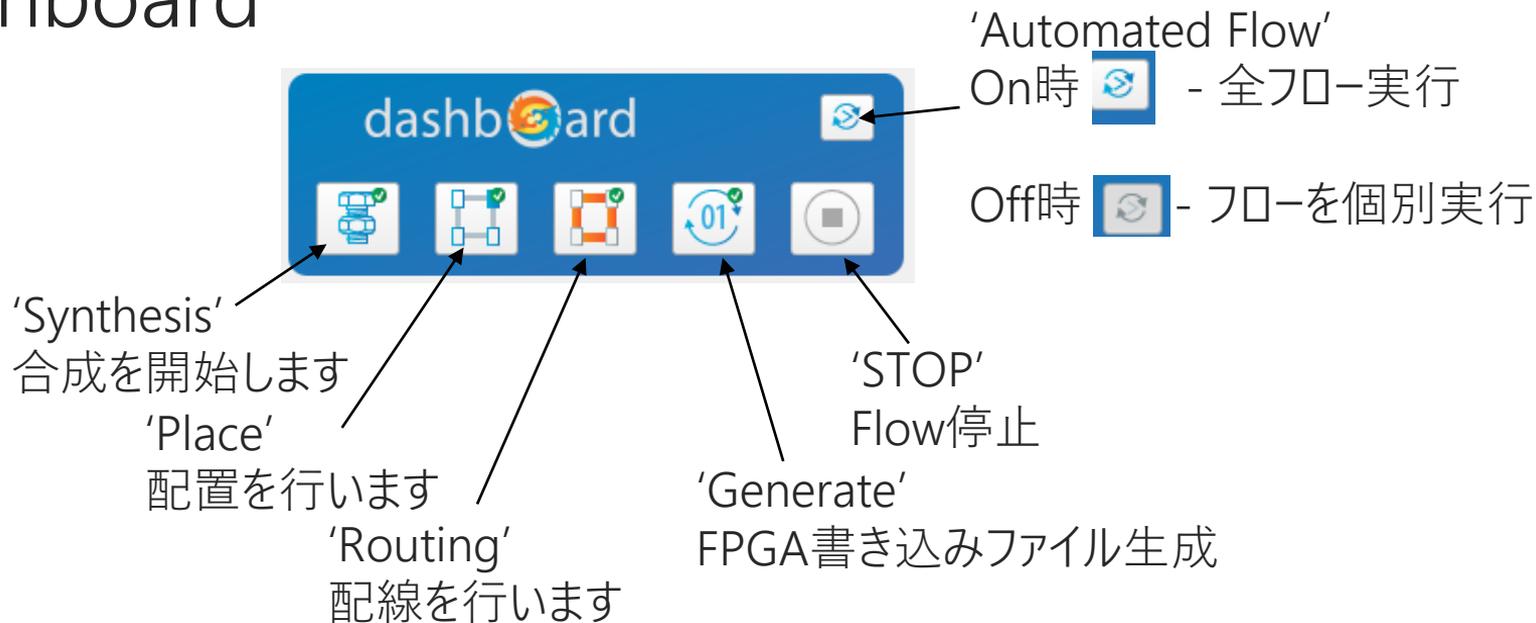


Tool Bar – 説明

- Exit - Efinityを終了する
- Preference - 基本設定
- Create Project - 新プロジェクト作成
- Open Project - 既存のプロジェクトを開く
- Edit Project - プロジェクト編集
- Close Project - プロジェクトを閉じる
- Show/Hide Console - Console画面表示
/非表示切り替え
- Load Place and Route Data
- Place and Route
結果表示
- Show Dashboard - Dashboard表示
- View Floor Plan - Floor Plan表示
- View Message Browser - Message Browser
表示
- View Timing Browser - Timing Browser表示
- View Log Message - Log表示
- Open Interface Designer
- Interface Designerを開く
- Open Debugger Wizard
- Debugger Wizard開始
- Open Debugger - Debugger起動
- Open Programmer - Programmer起動
- Open IP Catalog - IP選択 & 生成
- Show/Hide Tcl Command Console
- Tcl Command Console
表示/非表示切り替え



dashboard



フローで行われる処理

Automated FlowがOnの場合は以下の処理を実行

- ✓ デザインのシンタックス・チェック
 - ✓ 論理合成 (Synthesis)
 - ✓ 配置配線 (Place & Routing)
 - ✓ プログラミング・データ生成
 - ✓ タイミング検証
- ◆ 各プロセスの実行結果レポートが出力される

デザインを始める前に

- サポートされるHDL（ハードウェア記述言語）
 - ✓ Verilog HDL
 - Verilog 2001, Verilog 1995
 - VHDLとの混在可能
 - ✓ System Verilog
 - Verilog IEEE 1800 (System Verilog 2009,2005)
 - ✓ VHDL
 - VHDL IEEE 1076 (VHDL 2008,1993)
 - System Verilogとの混在可能
- System CなどのC言語は非サポート
- 回路図によるデザインエントリ機能は実装されていない

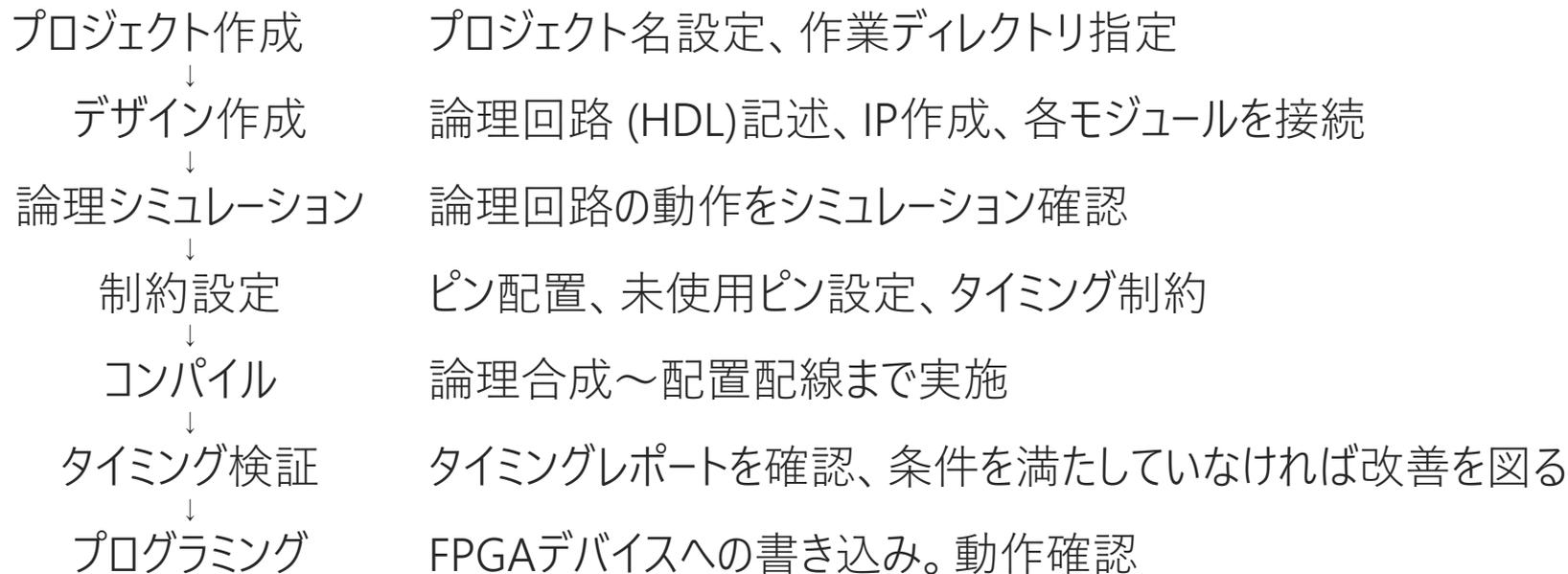


EfinityでのFPGAデザイン

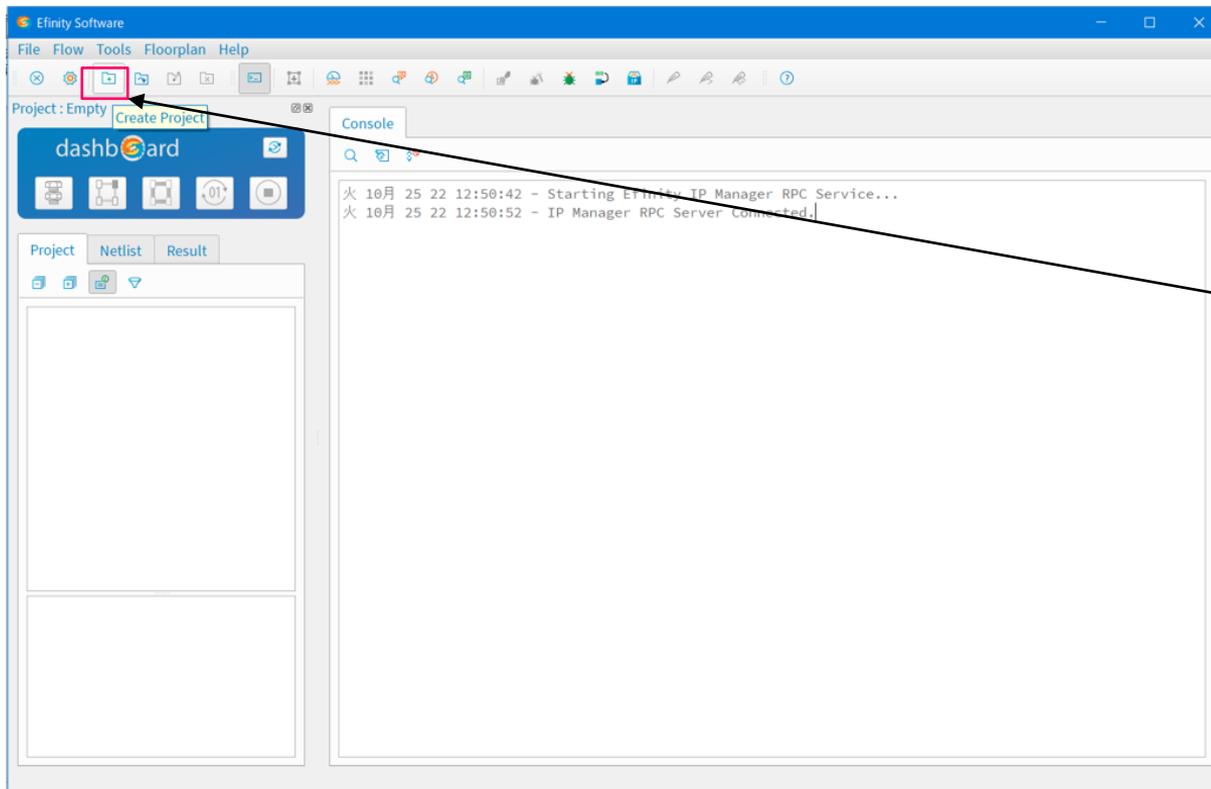
- 1つのFPGAデバイス開発に関わるすべてのファイルを「プロジェクト」で管理
 - 使用するすべての論理回路（デザイン）ファイル
 - ピン配置、制約ファイルなど
 - プロセス実行による生成ファイル
- 異なる（別の）FPGAデバイス開発には別プロジェクトを作成
- プロジェクト名と最上位階層デザインを一致させる（推奨）
- 作業を再開するときは既存のプロジェクトを開く



Efinityデザインフロー

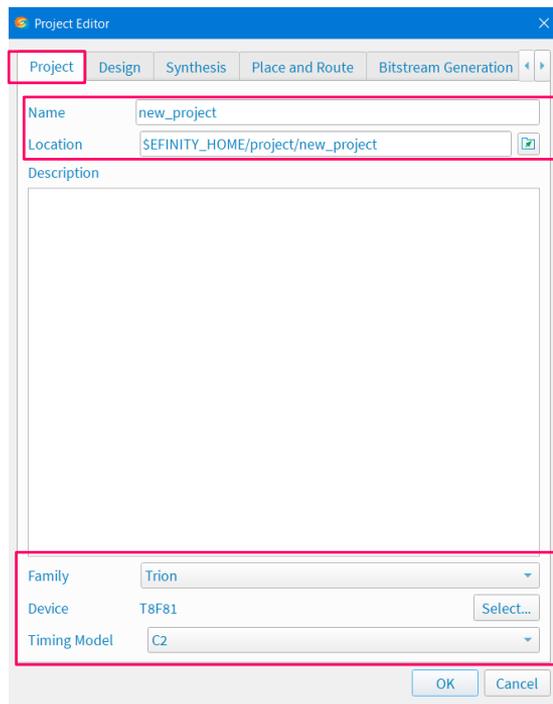


プロジェクト作成



‘Create Project’アイコン
をクリック。
プロジェクトを作成

プロジェクト作成ー続き



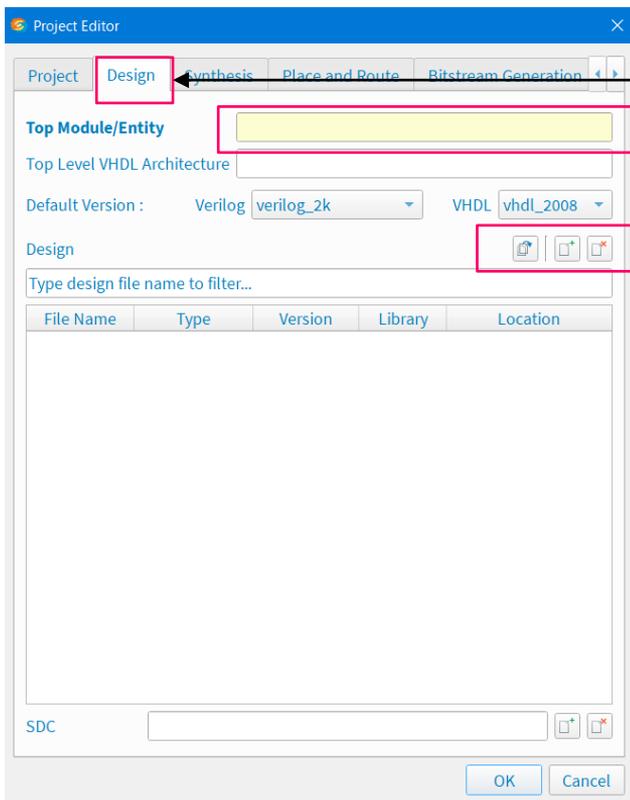
‘Create Project’を選ぶと
Project Editor画面が現れる。

‘Project’タブで

- プロジェクト名
 - 作業フォルダ
 - ターゲットデバイス
- を設定

Device Family (TrionまたはTitanium)
Device
Target Modelを選択

Designエントリー



'Design'タブでは

Top Moduleを指定

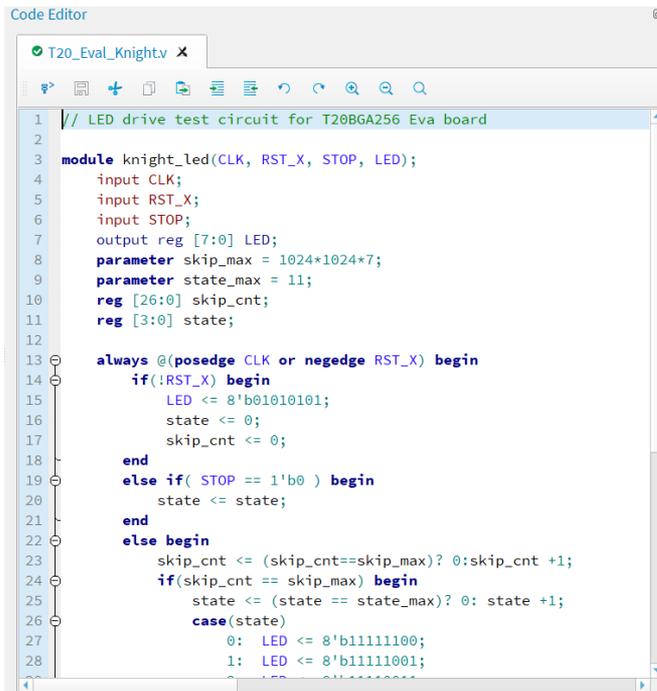
デザインファイルの

- 取り込み
- 追加
- 削除

を行うことができる

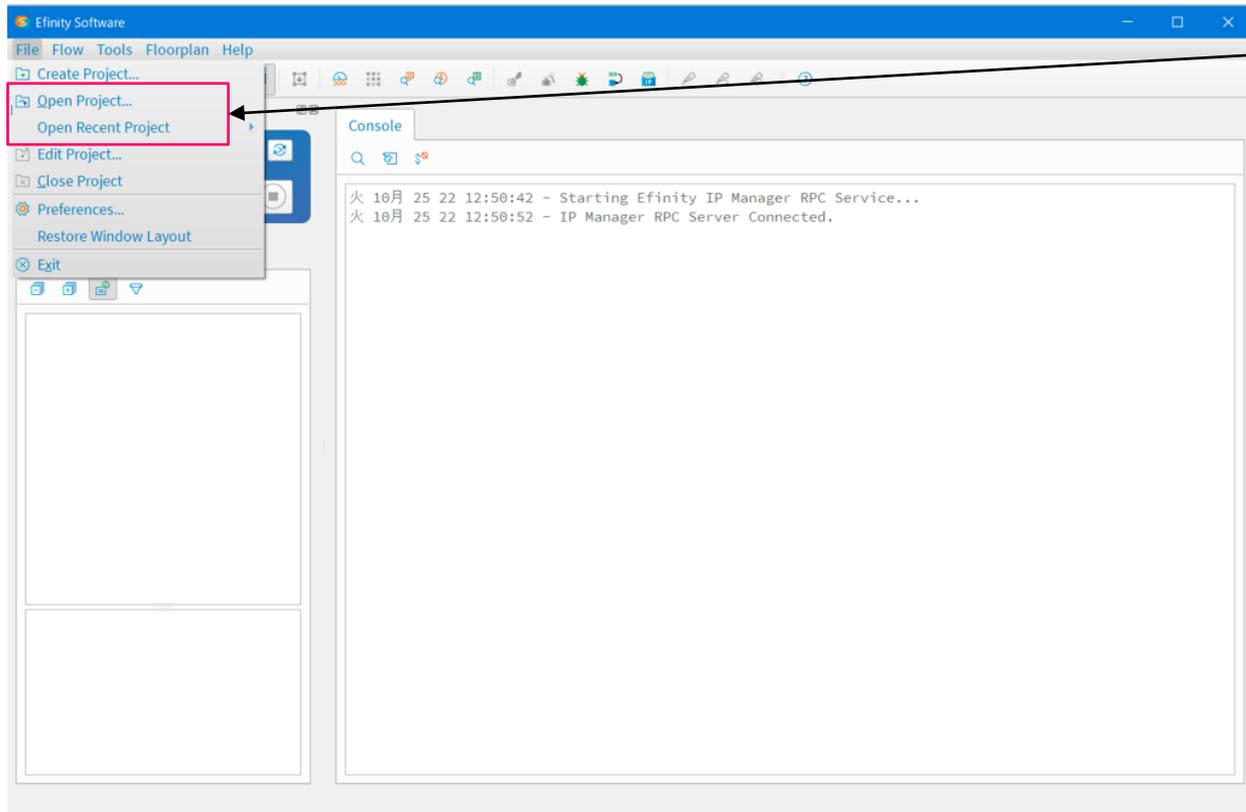
デザイン入力（作成）

- デザイン入力方式
 - Verilog HDL, VHDL, SystemVerilog
 - IP catalog
- Efinityには右記のようなエディタ機能も搭載



```
Code Editor
T20_Eval_Knight.v x
1 // LED drive test circuit for T208GA256 Eva board
2
3 module knight_led(CLK, RST_X, STOP, LED);
4     input CLK;
5     input RST_X;
6     input STOP;
7     output reg [7:0] LED;
8     parameter skip_max = 1024*1024*7;
9     parameter state_max = 11;
10    reg [26:0] skip_cnt;
11    reg [3:0] state;
12
13    always @(posedge CLK or negedge RST_X) begin
14        if(!RST_X) begin
15            LED <= 8'b01010101;
16            state <= 0;
17            skip_cnt <= 0;
18        end
19        else if( STOP == 1'b0 ) begin
20            state <= state;
21        end
22        else begin
23            skip_cnt <= (skip_cnt==skip_max)? 0:skip_cnt +1;
24            if(skip_cnt == skip_max) begin
25                state <= (state == state_max)? 0: state +1;
26                case(state)
27                    0: LED <= 8'b11111100;
28                    1: LED <= 8'b11111001;
```

既存プロジェクトを開く



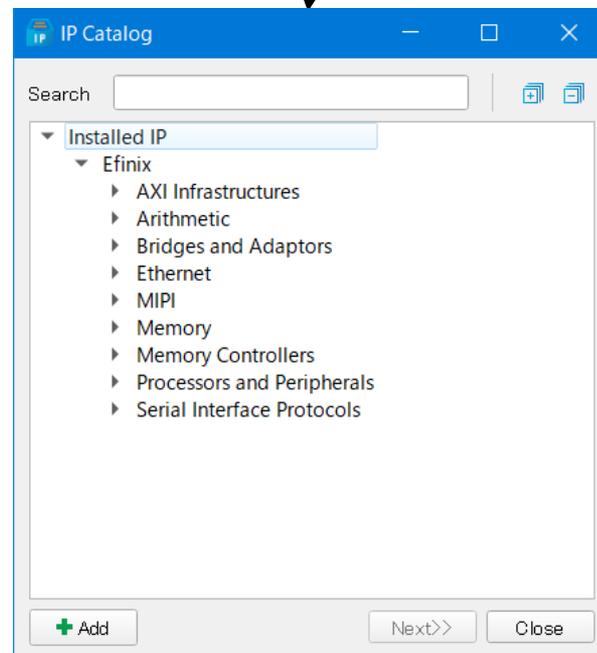
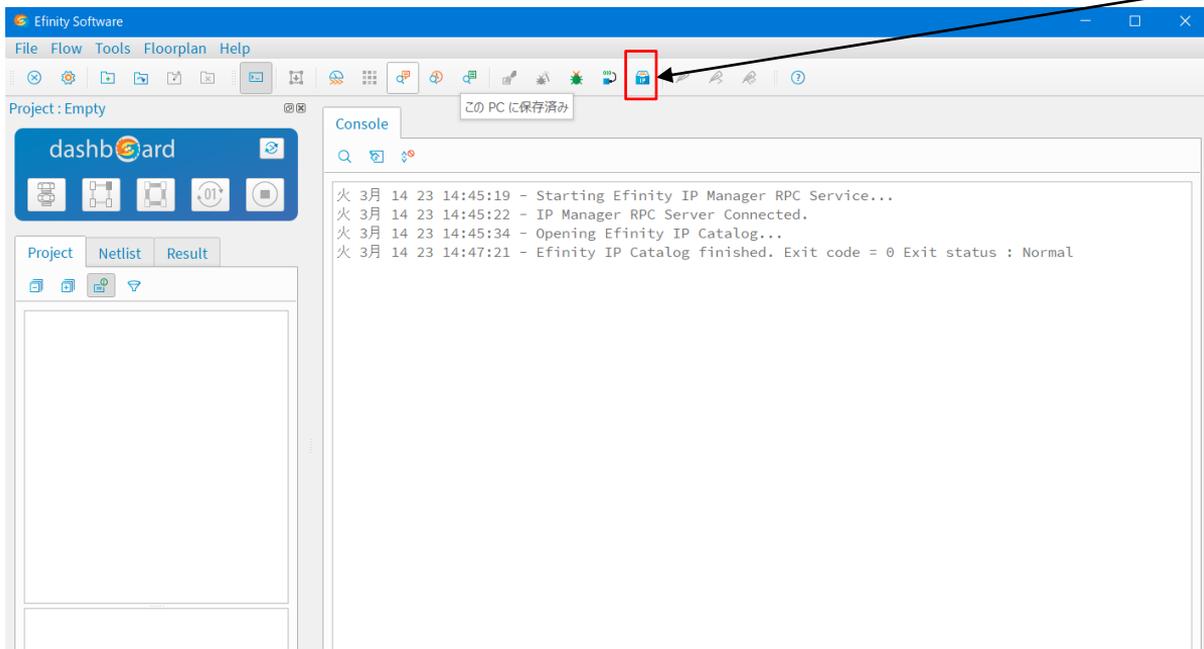
デザイン中のProjectを以下の手法で開くことができます

‘Open Project’は
ディレクトリ指定→
既存プロジェクトopen

‘Open Recent Project’は
最近開いたプロジェクトを
open

IP Catalog

IP Catalogアイコンを押すとIP Catalog画面が現れる



IP Catalog – 続き

- Efinix FPGA固有のIPライブラリ
 - ◆ RISC-V, MIPI, Memoryコントローラなど用意
 - ◆ 現状I社、X社と比較して提供されるIPは少ないが、今後随時追加される予定
- ゼロからの設計不要。ウィザード形式
 - ◆ パラメータをカスタマイズ。望みの形式にカスタマイズ
 - ◆ IP Catalogから所望のIPを選択。パラメータ設定 -> Generateで生成



Interface Designer



Interface Designerアイコン

Interface designerはEfinix社FPGAをデザインする上で

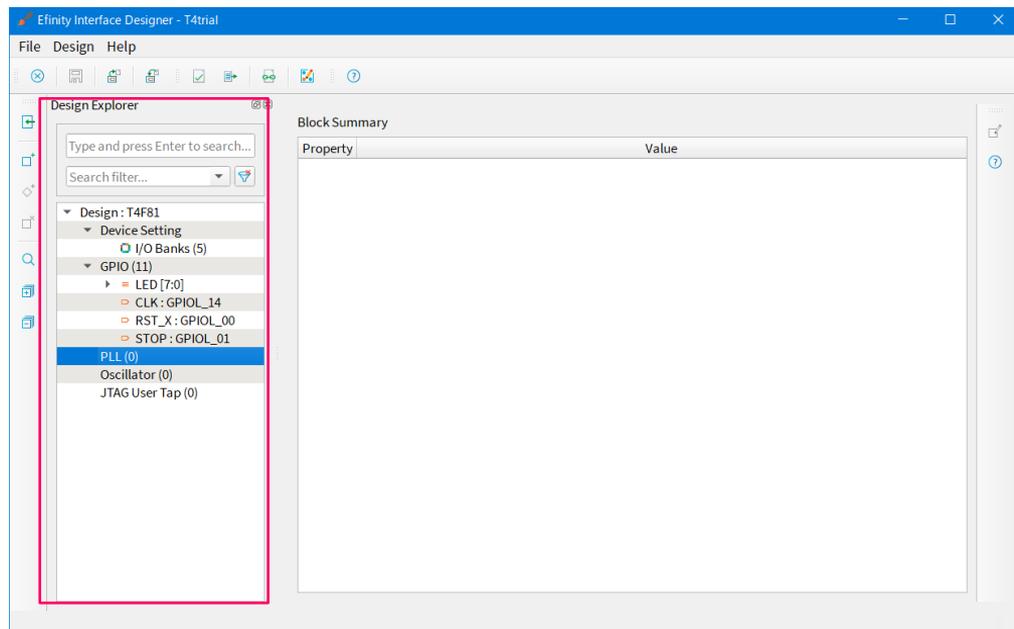
- GPIO
- PLL
- Oscillator
- MIPI
- LVDS
- JTAG User Tap
- DDR

などのInterfaceの定義・設定などを行うEfinityに搭載されている機能です。

Interface Designer - PIN配置指定

FPGAの合成を行うタイミングでは入出力（インターフェース）信号が定義されピン配置も決定されていることが必要条件となります。

▶ インターフェース信号を定義するためにInterface Designerが用意されています



Interface designer

- PINアサイン
 - I/O定義
 - Drive Strength
 - 未使用ピン定義
- などを行う。

状態は'Design Explorer'で確認できます

Interface Designer – GPIO Instance View

- Show/Hide GPIO Resource Assigner
- Create Block
- Create GPIO Bus
- Delete Block
- Show/Hide Search
- Expand All
- Collapse All

The screenshot shows the Efinity Interface Designer - T4trial window. The main area is titled "GPIO: Instance View" and contains a table with the following columns: Instance, Package Pin, Resource, I/O Bank, Alt Conn, Features, Clock Region, and Pad. The table is currently empty. To the left of the table is a toolbar with icons for Show/Hide GPIO Resource Assigner, Create Block, Create GPIO Bus, Delete Block, Show/Hide Search, Expand All, and Collapse All. Below the table is the Design Explorer, which has a search bar and a list of signals: GPIO (0), PLL (0), Oscillator (0), and JTAG User Tap (0). To the right of the Design Explorer is the Block Summary, which has two entries: 1 Global Unused Setting and 2 State input with weak pullup. To the right of the Block Summary is the Block Editor, which has a dropdown menu for Unused State with the value input with weak pullup.

Instance	Package Pin	Resource	I/O Bank	Alt Conn	Features	Clock Region	Pad
----------	-------------	----------	----------	----------	----------	--------------	-----

Property	Value
1 Global Unused Setting	
2 State	input with weak pullup

Unused State
input with weak pullup

Design Explorer

アサインする信号
を選択

未使用ピン定義

合成、プログラミング

デザインが完了したら合成を行います。



Automated Flowがアクティブな状態であれば
 'Synthesis'アイコンをクリックすることで全
フローが実行される。

左側に配置されるアイコンが優先であり例えば
 'Place'アイコンをクリックすれば、それ以
降のフローが実施される。

Automated FlowをOFFにすれば個々のフロー
を単独で実施可能

フロー終了

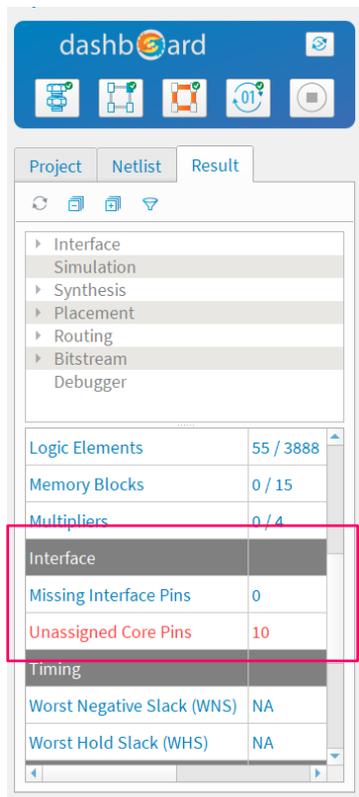
フローを実施すると自動的にResultタブに切り替わります。

The screenshot displays the Efinity Software interface. On the left, the 'Project' tab is active, showing a 'dashbord' with various icons and a 'Periphery Resource' table. A red box highlights the 'Project' and 'Result' tabs, with an arrow pointing from the 'Result' tab to the 'Console' window. The 'Console' window shows a log of system events, including 'Flow data refreshed', 'Project loaded VM', and 'Opening Efinity IP Catalog'. The 'Code Editor' window on the right shows a Verilog code snippet for a module named 'T20_Eval_Knight.v'. The code includes an 'else if' block for 'STOP == 1'b0', an 'else' block for 'skip_cnt' logic, and a 'case' statement for 'state' with 12 LED assignments and a default case.

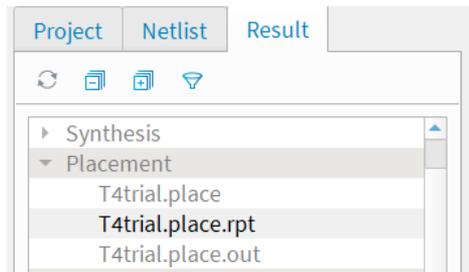
Periphery Resource	
GPIO	11 / 55
JTAG User TAP	0 / 2
Oscillator	0 / 1
PLL	0 / 1
Core Resources	
Inputs	3 / 96
Outputs	8 / 113
Clocks	1 / 16

```
19 else if( STOP == 1'b0 ) begin
20     state <= state;
21 end
22 else begin
23     skip_cnt <= (skip_cnt==skip_max)? 0:skip_cnt +1;
24     if(skip_cnt == skip_max) begin
25         state <= (state == state_max)? 0: state +1;
26         case(state)
27             0: LED <= 8'b11111100;
28             1: LED <= 8'b11111001;
29             2: LED <= 8'b11110011;
30             3: LED <= 8'b11100111;
31             4: LED <= 8'b11001111;
32             5: LED <= 8'b10011111;
33             6: LED <= 8'b00111111;
34             7: LED <= 8'b00111111;
35             8: LED <= 8'b10011111;
36             9: LED <= 8'b11001111;
37             10: LED <= 8'b11110011;
38             11: LED <= 8'b11111001;
39             default: LED <= 8'b00000000;
40         endcase
41     end
42 end
43 end
44 endmodule
45
46
```

エラーが出たら



エラー箇所は
赤字で表示



例えば左図のようなピンアサインエラーならば
'Placement' -> 'xxxx.place.rpt'
を確認する。 ^^^^ (プロジェクト名)
(上図参照)

◆ エラーに応じたレポートを確認する！

デバイスへの書き込み（プログラミング）

タイミングが要求仕様を満足していることが確認出来たらデバイスへの書き込み（プログラミング）を行います。

書き込みを行うには、ツールバーの'Open Programmer'アイコンをクリックします。

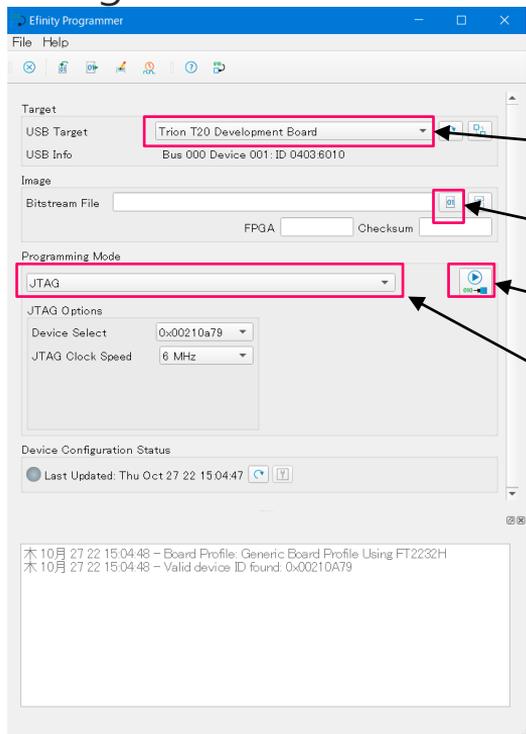


'Open Programmer'を押しプログラマーをopen

アイコンをクリックすると'Programmer'が現れます（次項）

Programmer

Programmer画面



Target Toolが表示される

書き込みファイル ('Bitstream') 選択

書き込み実行ボタン

書き込みモード選択

'JTAG'/'SPI Active'/'SPI Passive'/
'SPI Active using JTAG Bridge'/
'SPI Active x8 using JTAG Bridge'
の中から選択

詳細なデザイン設定・その他



続きの閲覧は、下記よりダウンロードお願いいたします。



https://www.restar-ele.com/business/electronic/efinix/efinix_efinity.html