

プロジェクト作成

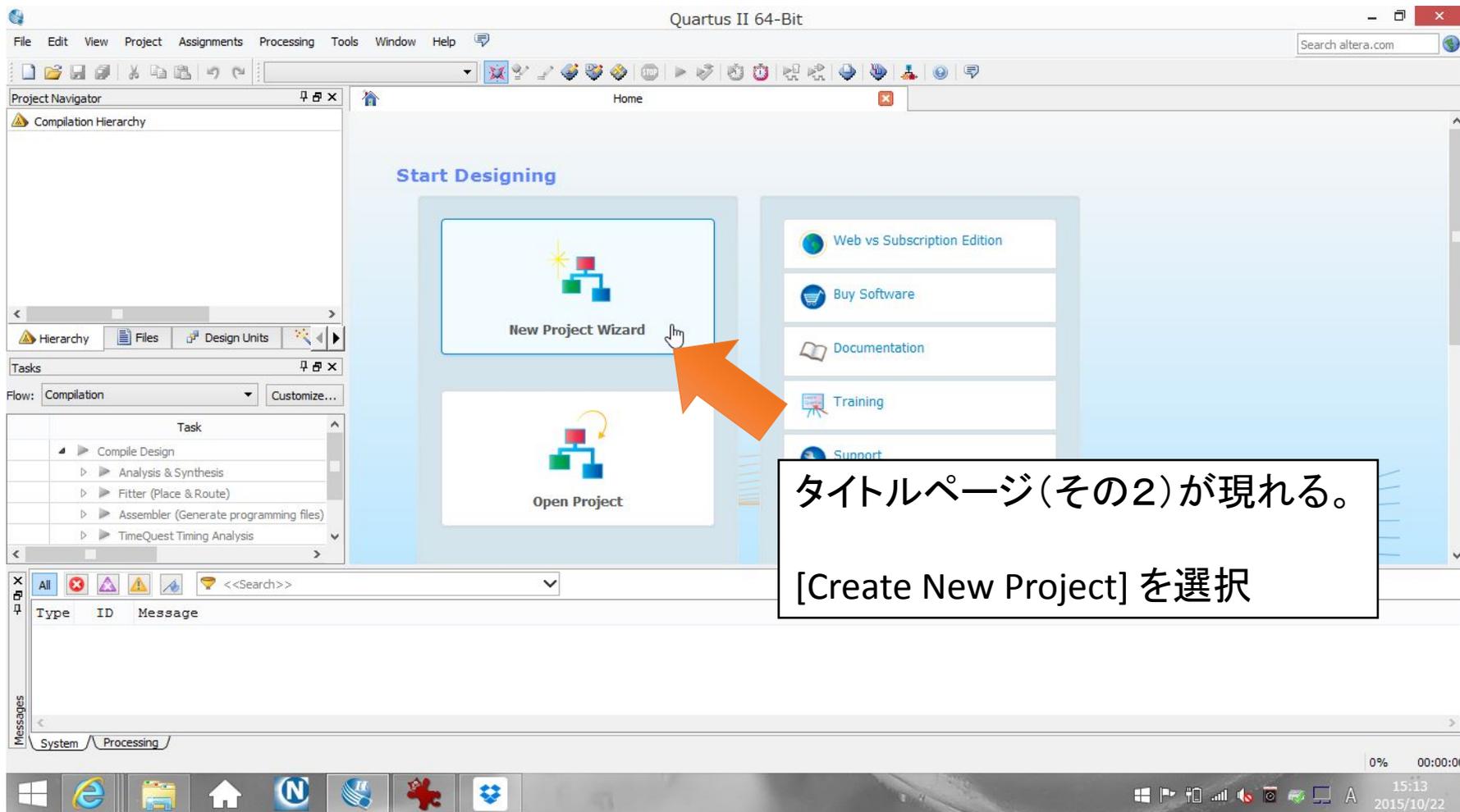
- Quartus IIを起動



タイトルページが現れる。

[Create New Project] を選択





プロジェクト名の決定

- プロジェクトという単位で開発を行います。
 - > ディレクトリ指定
 - > プロジェクト名の入力

トップレベルデザインを決定します
(自動で決定される。)

通常プロジェクト名と同じ名前で
モジュール名をつくり、そのモ
ジュールがトップレベルデザインと
なります。

今回は PwmCtrol という名前にしてみます。

New Project Wizard

Family & Device Settings [page 3 of 5]

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.

Device family

Family: Cyclone III

Devices: All

Target device

Auto device selected by the Fitter

Specific device selected in 'Available devices' list

Show in 'Available devices' list

Package: Any

Pin count: 484

Speed grade: 6

Name filter:

Show advanced devices

name	core voltage	LCS	user I/Os	memory bits	embedded multiplier 9-bit elements	PLL	bal Clo
EP3C16F484C6	1.2V	15408	347	516096	112	4	20
EP3C16U484C6	1.2V	15408	347	516096	112	4	20
EP3C40F484C6	1.2V	39600	332	1161216	252	4	20
EP3C40U484C6	1.2V	39600	332	1161216	252	4	20
EP3C55F484C6	1.2V	55856	328	2396160	312	4	20
EP3C55U484C6	1.2V	55856	328	2396160	312	4	20
EP3C80F484C6	1.2V	81264	296	2810880	488	4	20
EP3C80U484C6	1.2V	81264	296	2810880	488	4	20

< Back
Next >
Finish
Cancel
Help

小さい時ですが、
パッケージ上に
書かれています。



型番: EP3C16F484C6Nを選択する

大量にあるので、
デバイスファミリー、
ピン数、
スピードグレード
から絞込みを掛けたほう
がよいでしょう。

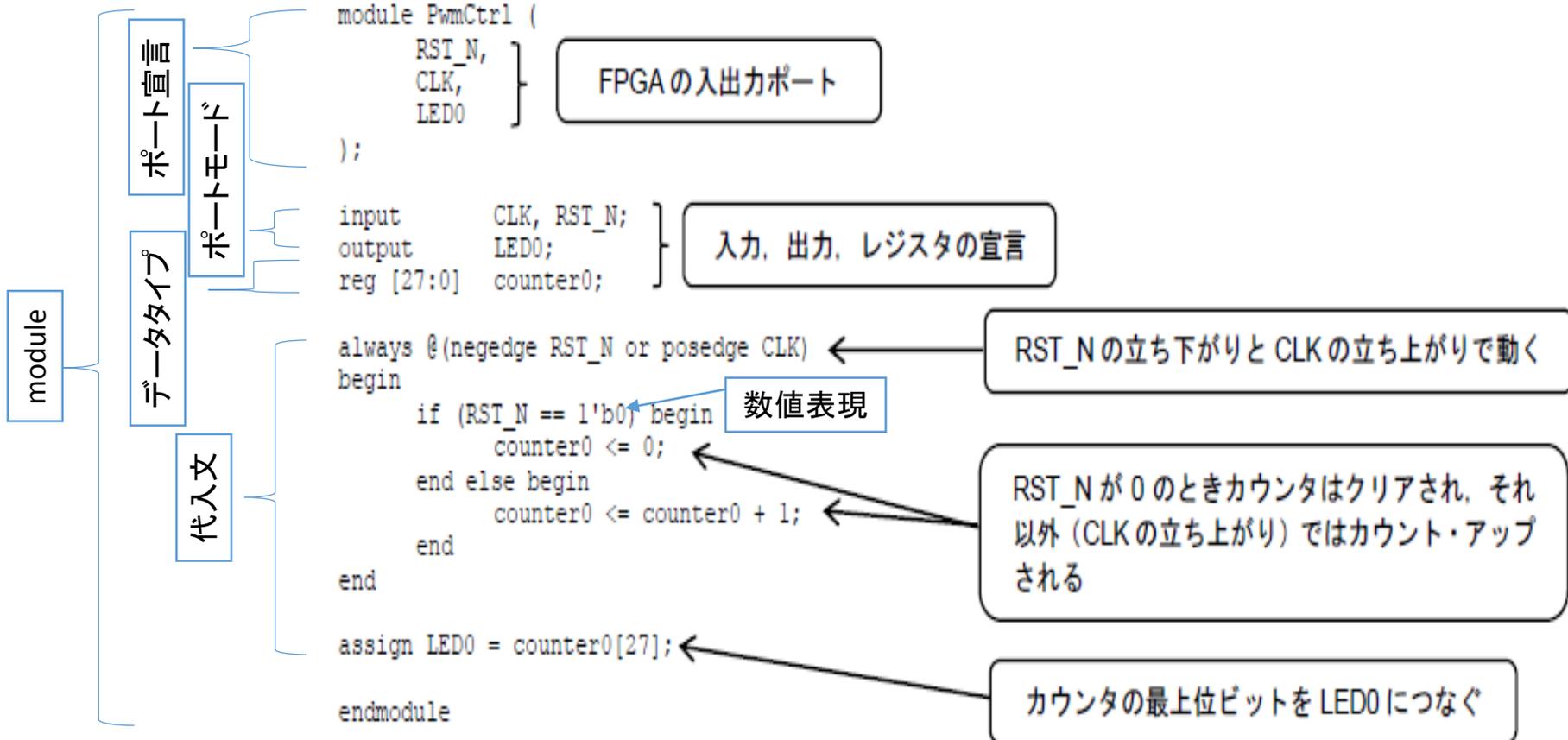
ソースを書く

デジタル回路を記述するには、回路図よりもハードウェア記述言語HDLの一種であるVerilog HDLを使う

- メニューからFile→Newを選ぶ
- ファイル名PwmCtrl.vとしてセーブ



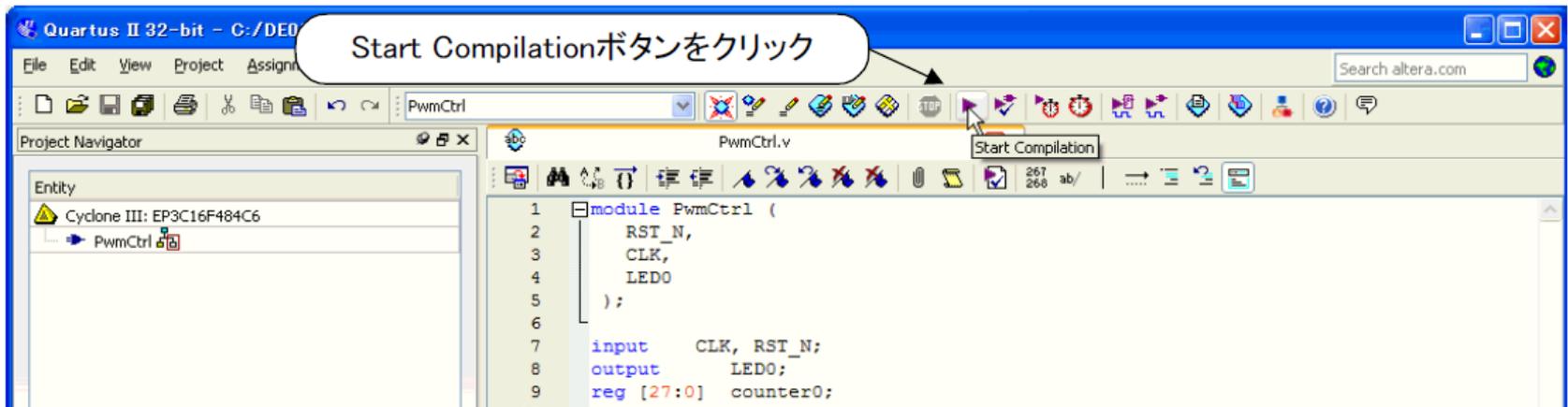
PwmCtrl.v のソースコード



一旦、コンパイル 「ソースコードのコンパイル」

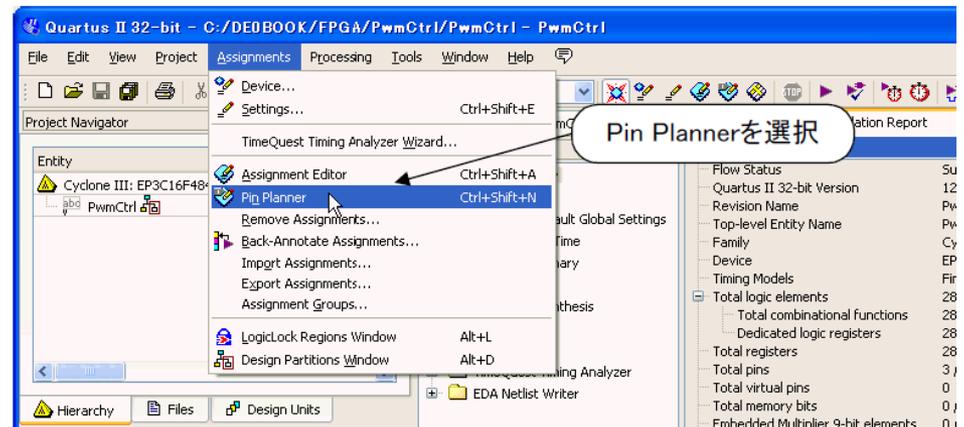
ソースコードをコンパイルします。

Quartus IIのメニューバーに「Start Compilation」のボタンがあるのでクリック
Successとできれば成功です。



ピンを割当てる

- メニューのAssignments→Pin Planner を選択



PIN PLANNER のマニュアル

https://www.altima.jp/members/japanese/download/pinassign_v10_1.pdf

Pin Planner

Pin Plannerが起動します。
さきほどのソースコードで
作成した CLK,LEDO,RST_N
が下の表に表示されています。

これを、FPGAのピンに割り当て
を行います。
実験で使用するFPGAは、484個
のピンがあります。
そのうちいくつかのPINの用途は
決まっていますが、他は自由に
つながることができます。

縦は A~AB まで22行あります
横は 1~22 まで22列あります。

ピンは行列であらわします。
例 右上「A22」

Top View - Wire Bond
Cyclone III - EP3C16F484C6

Node Name	Direction	Location	I/O Bank	VREF Group	Filter Location	I/O Standard	Reserved
CLK	Input				PIN_G2	2.5 V (default)	
LEDO	Output				PIN_E12	2.5 V (default)	
RST_N	Input				PIN_G1	2.5 V (default)	
<<new node>>							

信号は三つある。
入力2個と出力1個

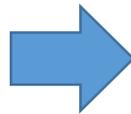
DE0 の回路図

DE0ボード上には、FPGAの他にLED、水晶発信機、プッシュボタンなど、多くの電子部品が配置されています。それらは、FPGAのピンに事前に接続されています。

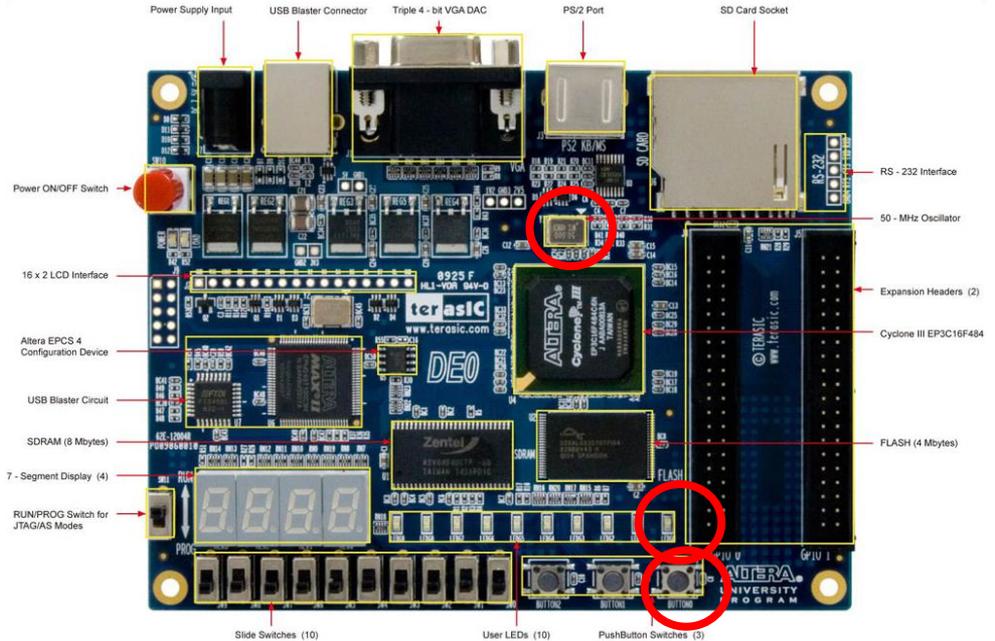


FPGAの「どのピン」が「どの外部回路」に接続しているかを知らなければなりません。

DE0_User_Manual.pdf



RST_N には 右下のプッシュボタン
CLK には水晶発信の信号
LED0 には右端のLED
を割り当てることにしましょう。



DE0 の回路図より

DE0_User_Manual.pdf

右下のプッシュボタンは H2
LED0は J1
水晶発信の信号は G21

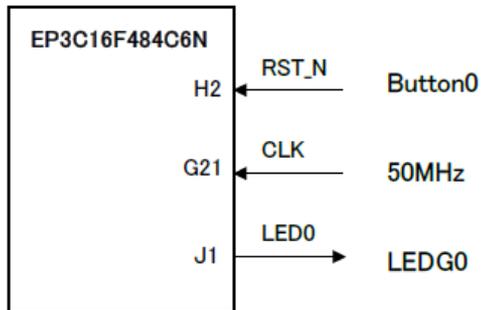


Table 4.2. Pin assignments for the pushbutton switches

Signal Name	FPGA Pin No.	Description
BUTTON [0]	PIN_H2	Pushbutton[0]
BUTTON [1]	PIN_G3	Pushbutton[1]
BUTTON [2]	PIN_F1	Pushbutton[2]

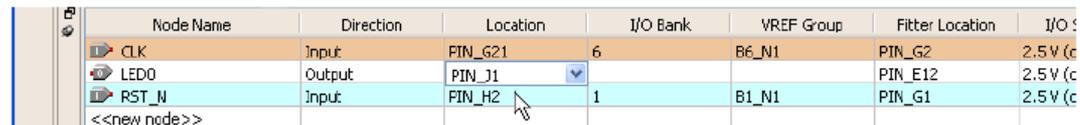
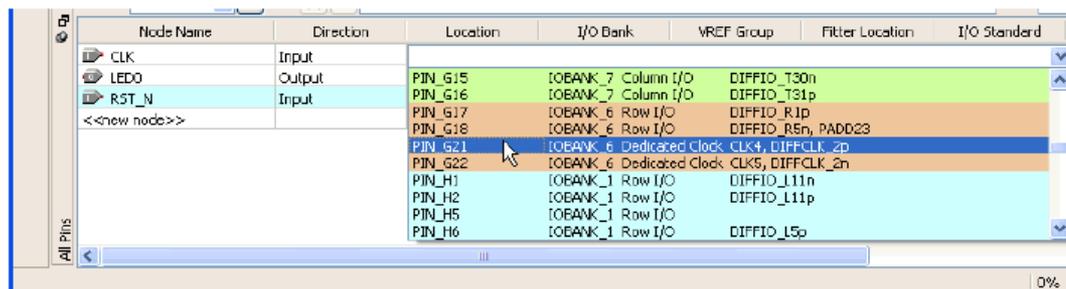
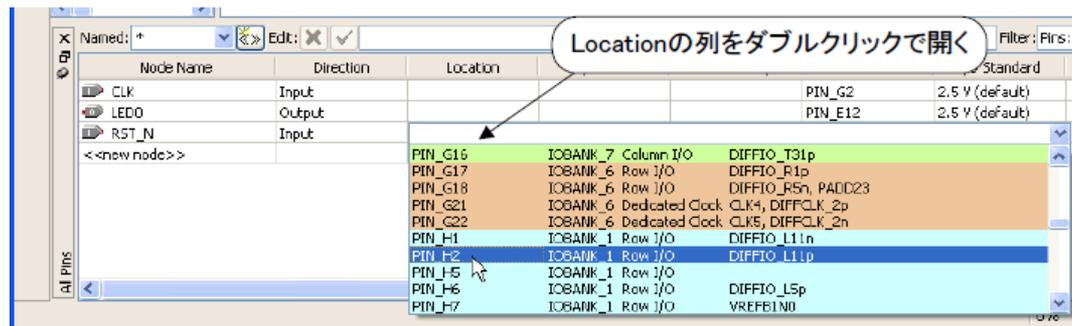
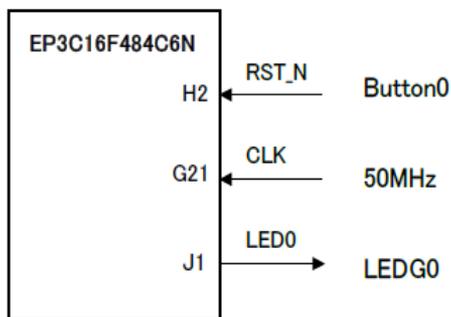
Table 4.3. Pin assignments for the LEDs

Signal Name	FPGA Pin No.	Description
LEDG[0]	PIN_J1	LED Green[0]
LEDG[1]	PIN_J2	LED Green[1]
LEDG[2]	PIN_J3	LED Green[2]
LEDG[3]	PIN_H1	LED Green[3]
LEDG[4]	PIN_F2	LED Green[4]
LEDG[5]	PIN_E1	LED Green[5]
LEDG[6]	PIN_C1	LED Green[6]
LEDG[7]	PIN_C2	LED Green[7]
LEDG[8]	PIN_B2	LED Green[8]
LEDG[9]	PIN_B1	LED Green[9]

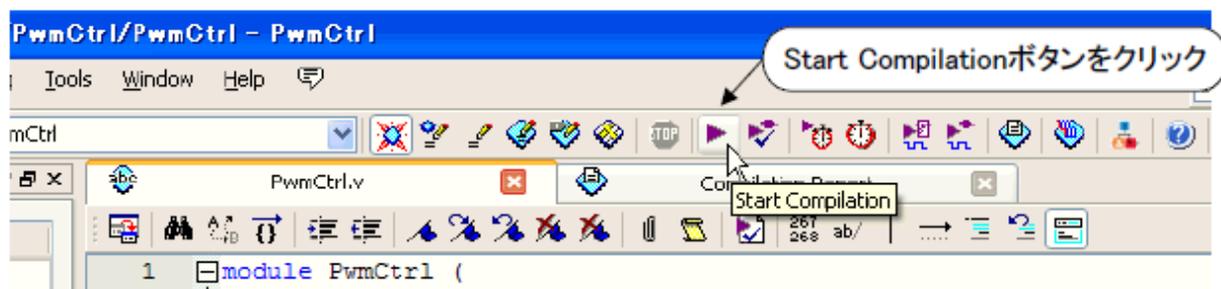
Table 4.5. Pin assignments for the clock inputs.

Signal Name	FPGA Pin No.	Description
CLOCK_50	PIN_G21	50 MHz clock input
CLOCK_50_2	PIN_B12	50 MHz clock input

Locationをクリックしてピンを割当てる



もう一度コンパイル



コンパイルによってVerilog HDL で書かれた回路はゲート・レベル (AND, OR, フリップフロップといった基本ハードウェア要素) の回路に変換されます(これを論理合成と言う)。



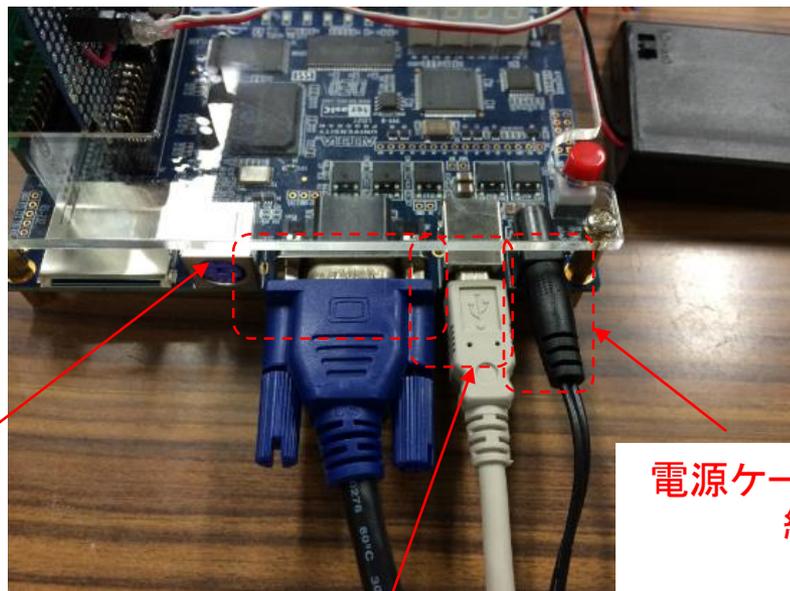
Device	EP3C16F484C6
Timing Models	Final
Total logic elements	28 / 15,408 (< 1 %)
Total combinational functions	28 / 15,408 (< 1 %)
Dedicated logic registers	28 / 15,408 (< 1 %)
Total registers	28
Total pins	3 / 347 (< 1 %)
Total virtual pins	0
Total memory bits	0 / 516,096 (0 %)
Embedded Multiplier 9-bit elements	0 / 112 (0 %)
Total PLLs	0 / 4 (0 %)

ロジカルエレメント(論理要素)LEは28個しか使っていない

347ピンのうち3つ使用した

DEO に電源を入れ，さらにパソコンと USB ケーブルでつなぎます。

- FPGAとPCの接続
- 実習用のPCには事前に USB-Blasterのドライバーがインストールされており、自動的にPCとFPGAとの接続が確立します。



液晶ディスプレイを接続
(カメラ画像の確認用)
VGAを用いた実習で必要。
今回は、接続しなくても良い。

開発用PCとUSBケーブルで接続
(プログラムインストール用)

FPGAを用いた組み込みシステムの協
調設計

電源ケーブルを接
続

FPGAスタンドア
ローンでの実行に
必要。今回は接続
しなくても可

1. 「Programmer」起動

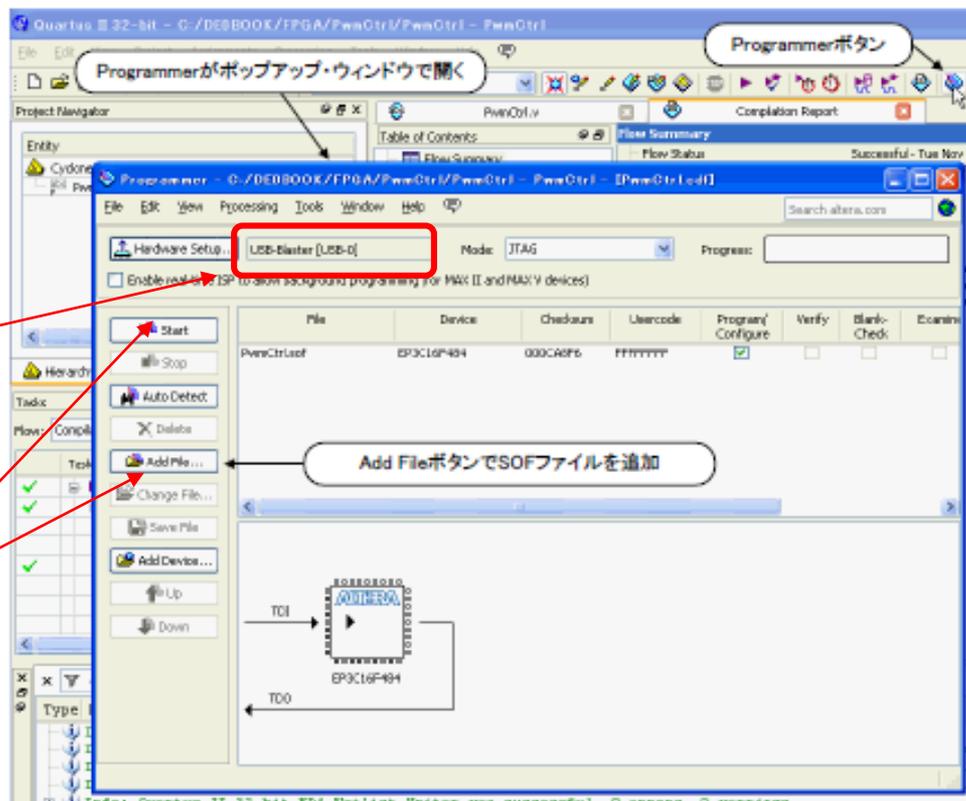
右上の  アイコンをクリック

2. 対象HWが登録を確認

USB-Blasterが選択されていることを確認。されていない場合「HardwareSetup」を押し、選択

3. SOFファイルの追加

4. SOFファイルの実行



実行結果

LED0が約5.37秒周期で点滅

```
module PwmCtrl (  
    RST_N,  
    CLK,  
    LED0  
);
```

```
input    CLK, RST_N;  
output  LED0;  
reg [27:0] counter0;
```

```
always @(negedge RST_N or posedge CLK)  
begin  
    if (RST_N == 1'b0) begin  
        counter0 <= 0;  
    end else begin  
        counter0 <= counter0 + 1;  
    end  
end
```

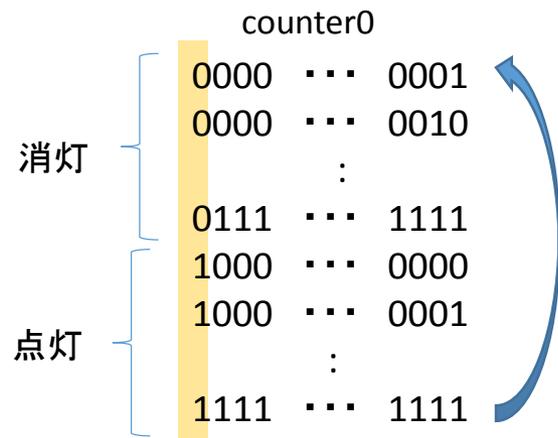
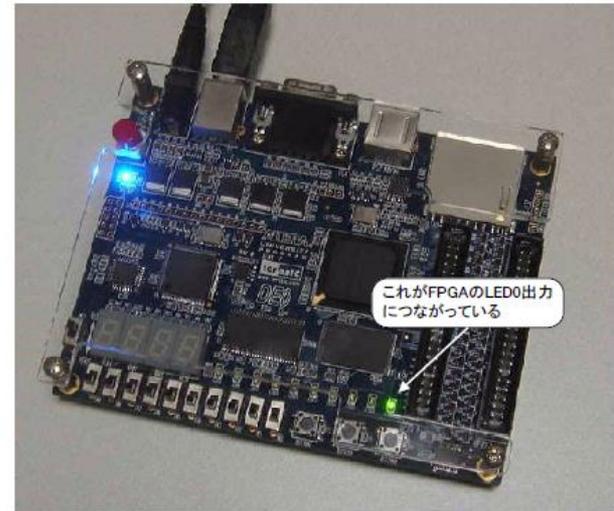
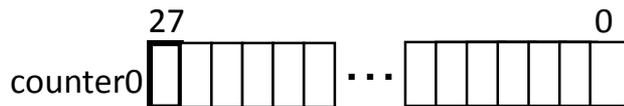
```
assign LED0 = counter0[27];  
endmodule
```

$2^{27} = 134,217,728$
水晶発信機は50MHzなので
 $2^{27} \div 50\text{MHz} = 2.68\text{秒}$

Count0はRST_Nが0のときの
CLKの立上りで0となる。

Count0はCLKの立上り
毎に+1加算される。

Count0の27ビットはLED0と接続されている。



LED0との接続は1でHiOでLowレベル出力電圧 => アクティブハイ回路

マイコンを用いた協調設計

- AND,OR,フリップフロップ、加算器、乗算器という基本ハードウェア要素をつなぎ合わせるだけでは、設計に限界がある。
- 柔軟でわかりやすいシステム設計には「マイコン」が便利

