

EDA-002 ユーザーズマニュアル

第 2 版

目次

はじめに.....	2
付属品.....	2
ご注意事項.....	2
概要.....	3
基板外観.....	3
ブロック図.....	3
開発環境.....	4
ダウンロードケーブル.....	4
クロック.....	4
7セグメント LED.....	4
押しボタンスイッチ.....	4
赤色 LED.....	4
電源.....	4
FPGA コンフィグレーション.....	4
FPGA ピンアサイン表.....	5
クロック.....	5
7セグメント LED.....	5
押しボタンスイッチ.....	5
7セグメント LED.....	5
USB ドライバのインストール.....	6
USB ドライバのアンインストール.....	8
Quartus II による開発手順.....	10
プロジェクトの新規作成.....	10
VHDL による AND 回路.....	13
新規ソースの追加.....	14
コンパイル.....	16
ピンの割り当て.....	17
シミュレーション.....	20
RBF ファイルの作成.....	26
専用ソフトウェアによる FPGA コンフィグレーション.....	29
動作の確認.....	30
EDA-002 サンプル回路.....	31
サンプル回路の動作.....	31
サンプル回路のコンフィグレーション手順.....	31
付属 CD-ROM の内容.....	32

はじめに

この度は、FPGA トレーナ EDA-002 をお買い上げいただきまして、誠にありがとうございます。

EDA-002 は、ALTERA 社の FPGA である ACEX (EP1K10TC100-3 : 56,000 ゲート) を実装したトレーナです。4桁数字表示 LED、汎用 LED、押しボタンスイッチ、クロックモジュール、FPGA コンフィグレーションに使用する USB インターフェースを実装しておりますので、快適に FPGA 設計を進めることができます。

VHDL や Verilog-HDL による論理回路設計の習得に、ご活用ください。

付属品

EDA - 002 本体
USB ケーブル
マニュアル(本書)
ユーザー登録はがき
付属 CD(P32 参照)

ご注意事項

本製品及び本書の内容は、改良のために予告なく変更することがあります。

本書の内容については万全の記して作成しましたが、万一誤りなど、お気づきの点がございましたら、ご連絡をお願いいたします。

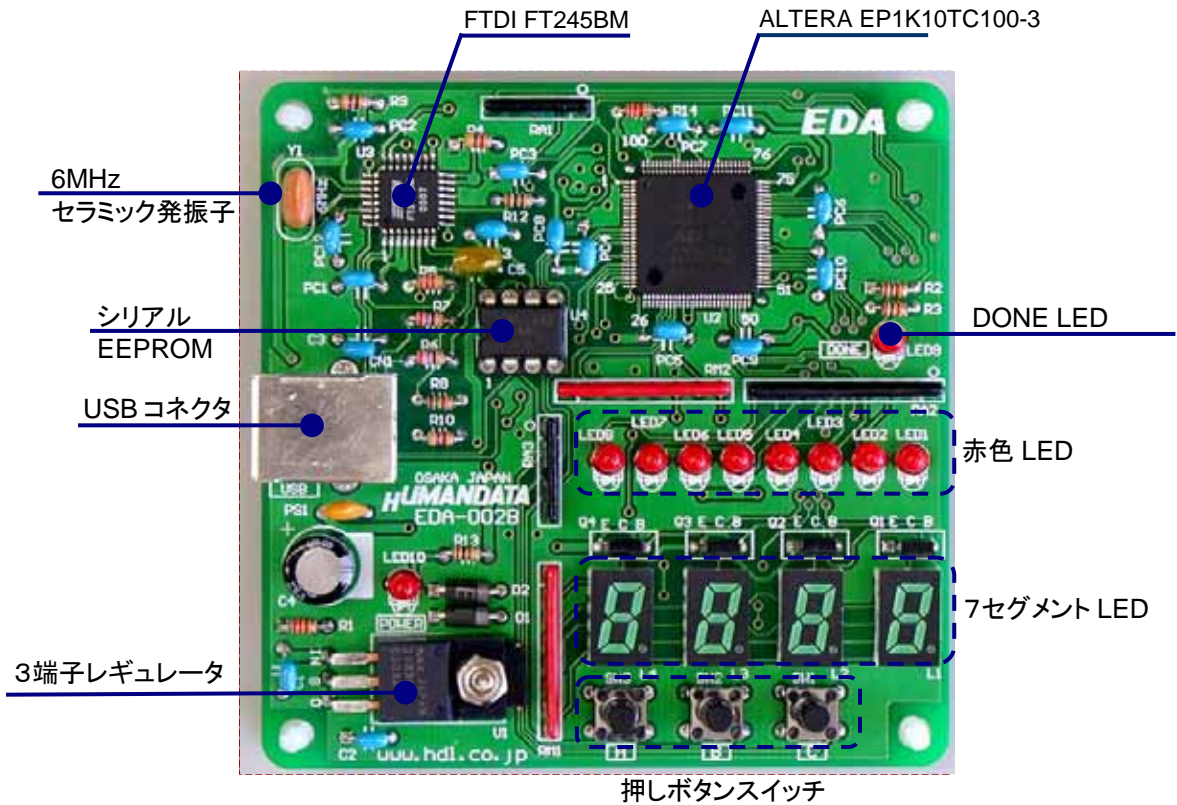
本製品の運用の結果につきましては、2. 項にかかわらず当社は責任を負いかねますので、ご了承願います。

本書に記載されている使用と異なる使用をされ、あるいは本書に記載されていない使用をされた場合の結果については、当社は責任を負いません。

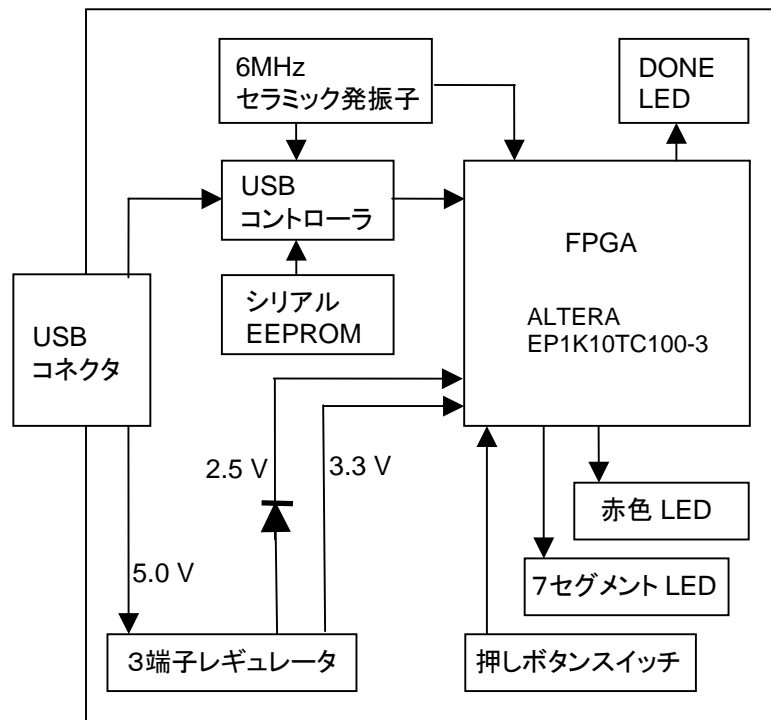
本書および、回路図、サンプル回路などを無断で複写、引用、配布することはお断りいたします。

概要

基板外観



ブロック図



開発環境

FPGAの内部回路設計には、回路図エディタやHDL入力ツール、論理合成ツールが必要です。これらの開発ツールは、ALTERA社が無償配布するQuartus IIにて可能です。使用する際には、インターネットによるライセンス登録が必要となります。

ダウンロードケーブル

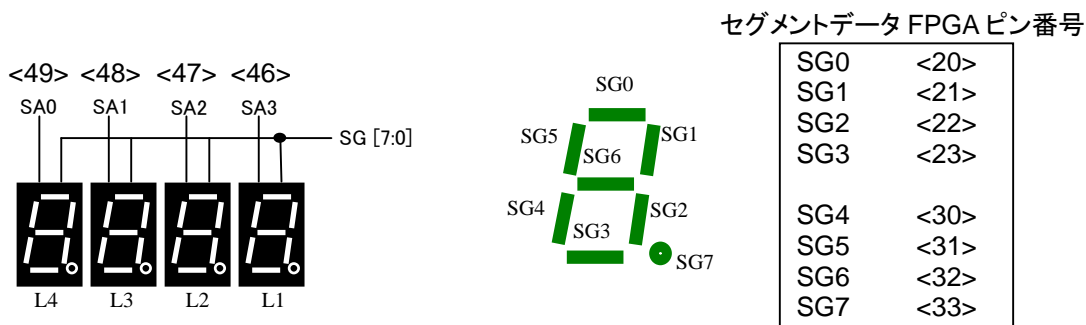
FPGA へのコンフィグレーションには、専用のダウンロードケーブルを必要としません。添付の USB ケーブルをご使用ください。

クロック

6MHz セラミック発振子より、USB コントローラと FPGA にクロックを供給します。

7セグメント LED

4 つの 7SEG の一つを選択する信号 SA0、SA1、SA2、SA3 のいずれかを L とし、その他をハイインピーダンスまたはオープンドレインにてオフとします。数 kHz で順番に表示させることによって、連続して点灯しているようにみえます。また、複数を同時に L とすると電流が過大となりますので、注意して下さい。



押しボタンスイッチ

押すと L になります。このピンは FPGA に対して、入力となりますので、出力しないでください。 押しボタンスイッチに対して **FPGA 側から出力するとショートします**ので注意してください。

赤色 LED

赤色 LED は負論理です。Low にて点灯します。完全に消灯させるには、ハイインピーダンスとするか、オープンドレインにてオフとしてください。

電源

電源は USB から 5.0V が供給されます。FPGA が必要とする 3.3V、2.5V はオンボードのレギュレータとダイオードにより生成されます。

FPGA コンフィグレーション

FTDI 社提供のダイレクトドライバをインストールすることで、USB 経由で FPGA をコンフィグレーションすることができます。コンフィグレーションが終了し、DONE 信号が H になると 赤色 LED (LED9) が点灯します。

FPGA ピンアサイン表

クロック

信号名	I/O	ピン番号	備考
CLK0	I	P39	6MHz

7セグメント LED

信号名	I/O	ピン番号	備考
SA0	O	P49	
SA1	O	P48	
SA2	O	P47	
SA3	O	P46	
SG0	O	P20	
SG1	O	P21	
SG2	O	P22	
SG3	O	P23	
SG4	O	P30	
SG5	O	P31	
SG6	O	P32	
SG7	O	P33	小数点

押しボタンスイッチ

信号名	I/O	ピン番号	備考
PSW_A	I	P15	
PSW_B	I	P14	
PSW_C	I	P13	

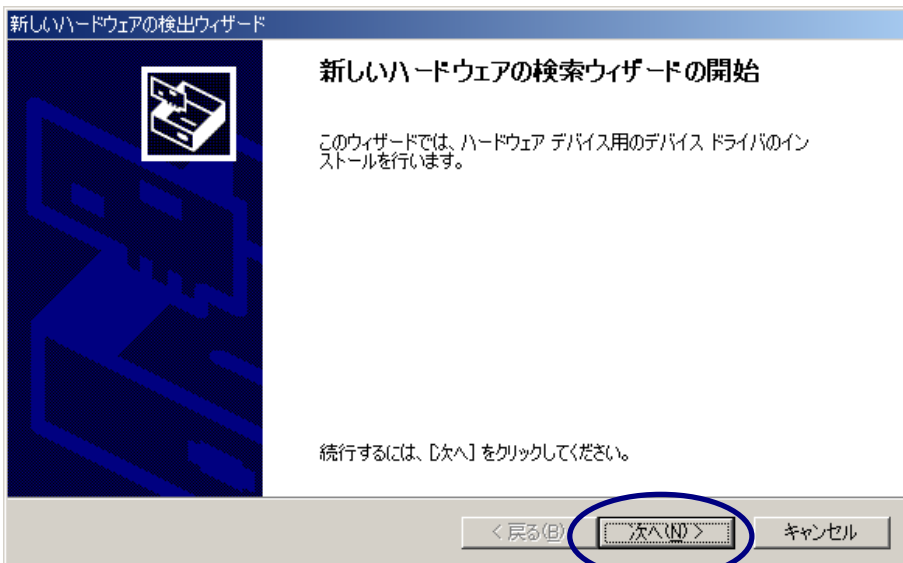
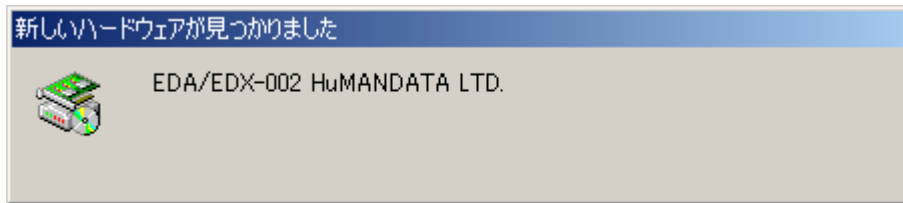
7セグメント LED

信号名	I/O	ピン番号	備考
LED0	O	P71	
LED1	O	P70	
LED2	O	P69	
LED3	O	P68	
LED4	O	P58	
LED5	O	P57	
LED6	O	P56	
LED7	O	P55	

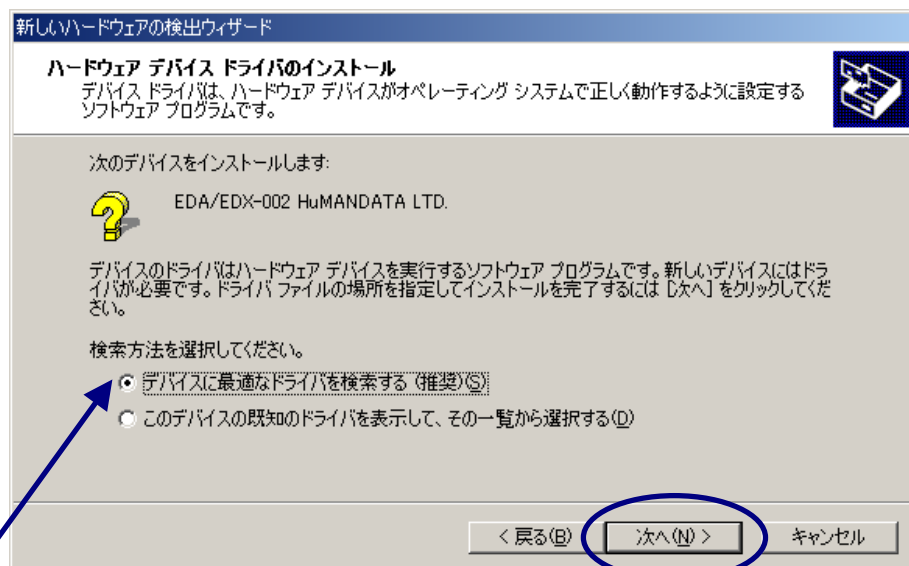
USB ドライバのインストール

FPGA へのコンフィグレーションと、USB の通信実験の前には、FTDI 社の提供するドライバを PC にインストールする必要があります。本章で説明するインストール作業が完了後、FPGA のコンフィグレーションが可能となります。

USB コネクタに付属 USB ケーブルを挿入すると、「新しいハードウェア」が認識され、次のダイアログが表示されます。次の手順に従ってインストール作業を完了ください。

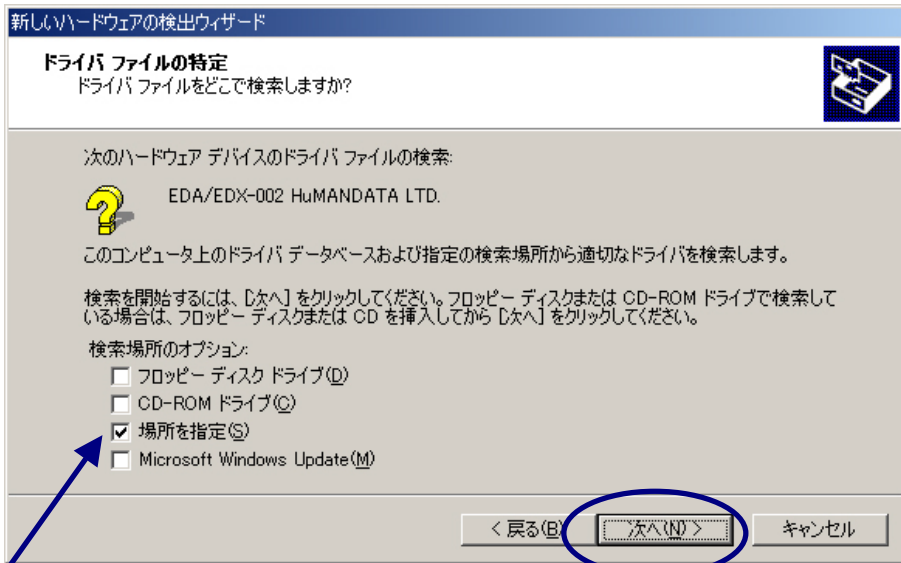


「次へ」をクリック



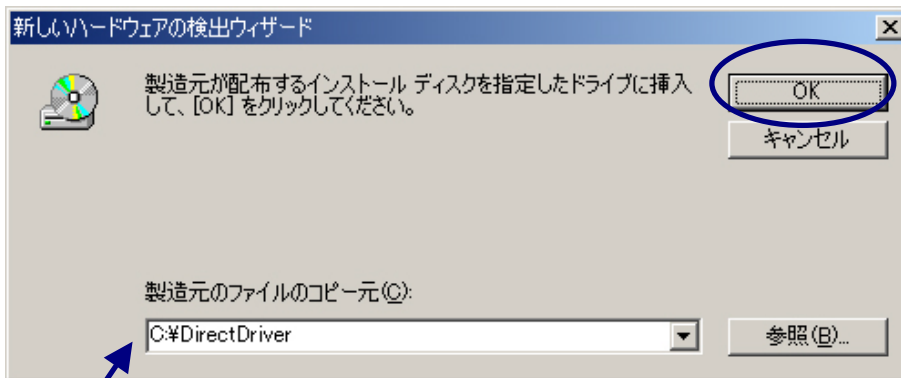
「次へ」をクリック

「デバイスに最適なドライバを検索する(推奨)」を選択



「場所を指定」を選択

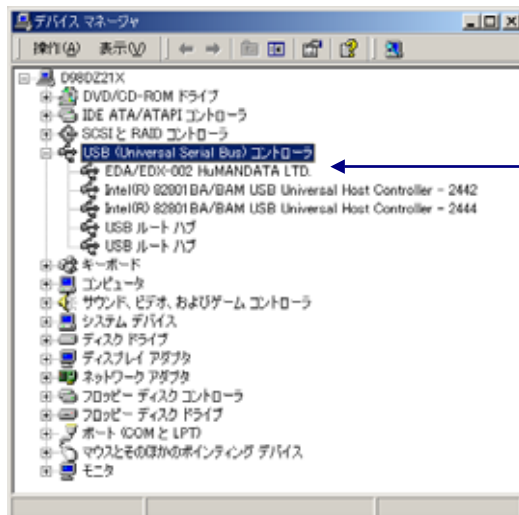
「次へ」をクリック



「OK」をクリック

「ftd2xx.inf」の格納場所を指定します。付属 CD の DirectDriver を選択して下さい

コントロールパネルのデバイスマネージャを開いてください。USB (Universal Serial Bus) コントローラとポートの項目に、「EDA / EDA-002 HuMANDATA LTD.」が追加されているか確認してください。



「EDA/EDA-002 HuMANDATA LTD.」

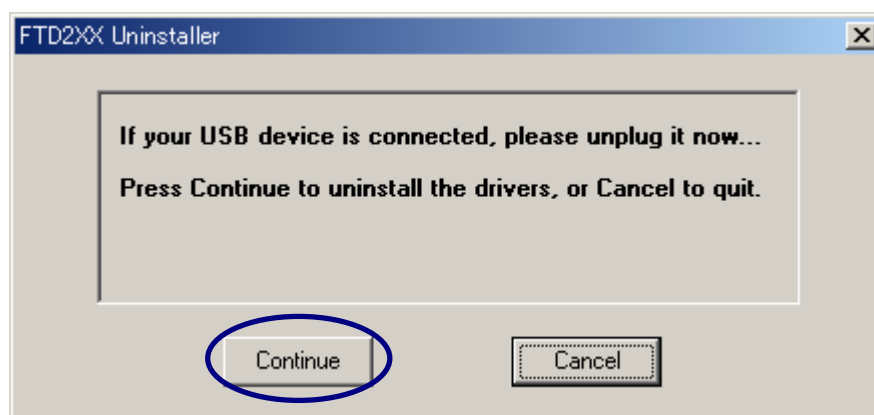
正常にインストールされた状態

USB ドライバのアンインストール

インストールした USB ドライバをアンインストールには、**PC と EDA-002 を接続しない状態で**「スタート」 - 「設定」 - 「コントロールパネル」 - 「アプリケーションの追加と削除」から「EDA/EDX-002 HuMANDATA LTD.」を選択し、「変更と削除」を実行してください。

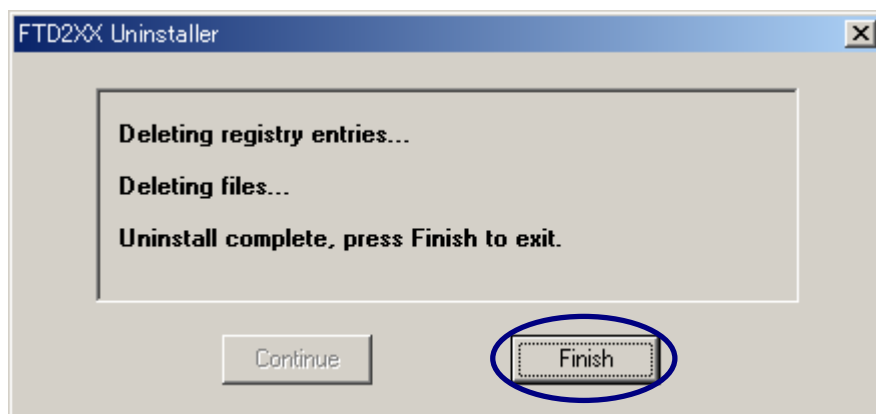


実行すると、次のようなダイアログが表示されます。



「Continue」をクリック

次のダイアログで「Finish」をクリックすれば、終了です。



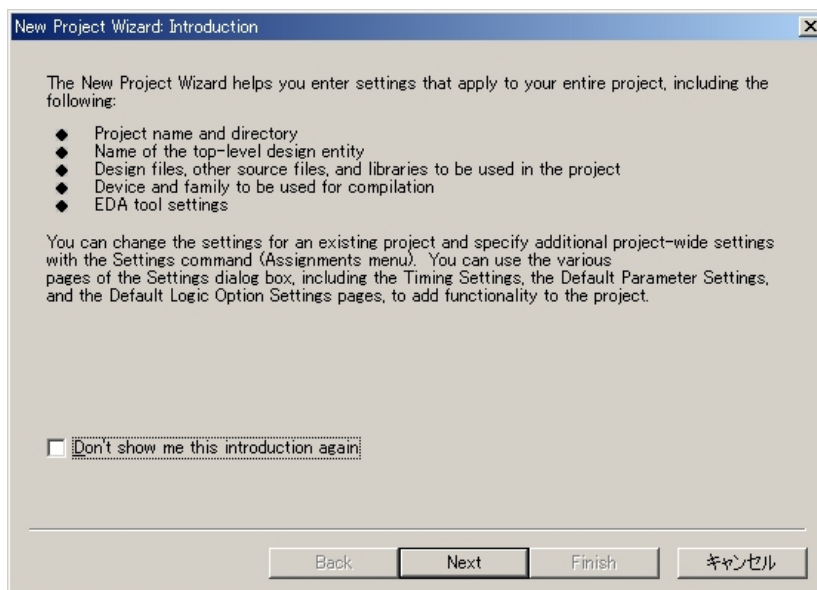
「Finish」をクリック

Quartus II による開発手順

EDA-002 に搭載されている FPGA をコンフィグレーションするには rbf ファイルが必要となります。このファイルを生成する手順について説明します。rbf ファイルを生成するには ALTERA 社の Quartus II が必要となります。

プロジェクトの新規作成

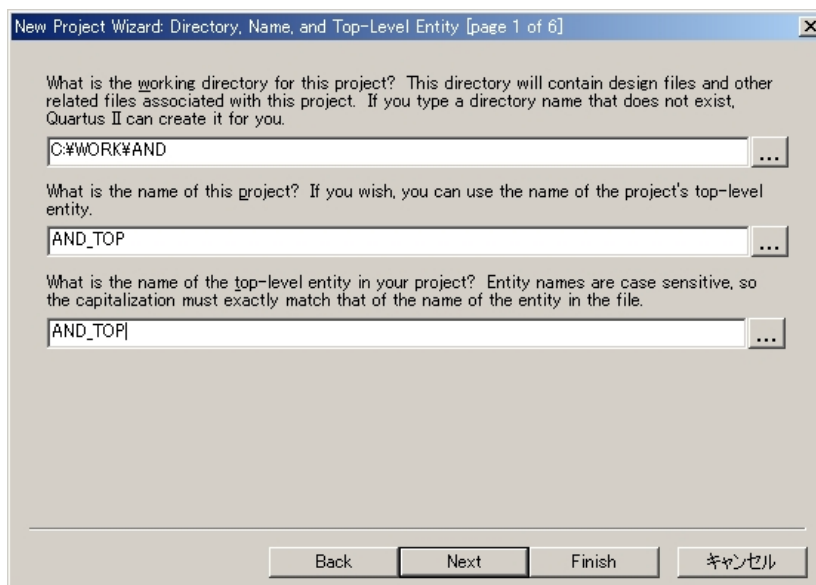
プロジェクトを新規作成するには、[File] [New Project Wizard] をクリックします。下の画面が、出るので「Next」をクリックします。



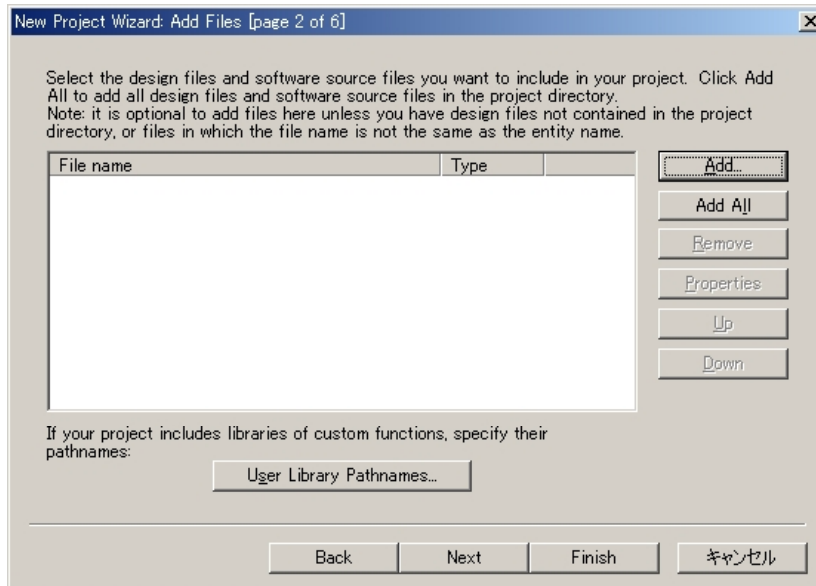
プロジェクトディレクトリは、「C: ¥WORK」とします。

上段	プロジェクトを保存するフォルダ	C: ¥WORK ¥AND
中断	プロジェクト名	AND_TOP
下段	作成するロジックの Entity 名	AND_TOP

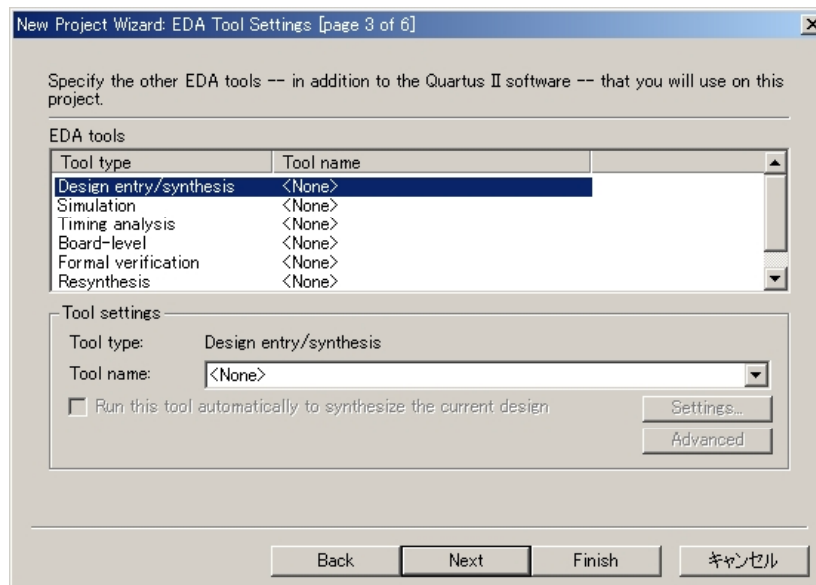
入力項目は、上記ですが、中断のプロジェクト名と下段の作成するロジックの Entity 名は、同じにしておかなければこの先のコンパイルでエラーになります。



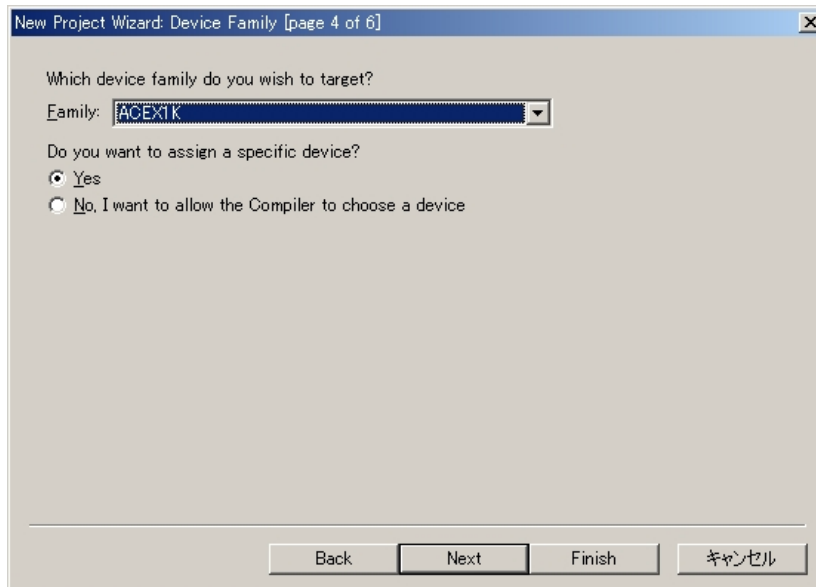
「Next」をクリックすると下の画面が、出ますが、何も入力せずに「Next」をクリックします。



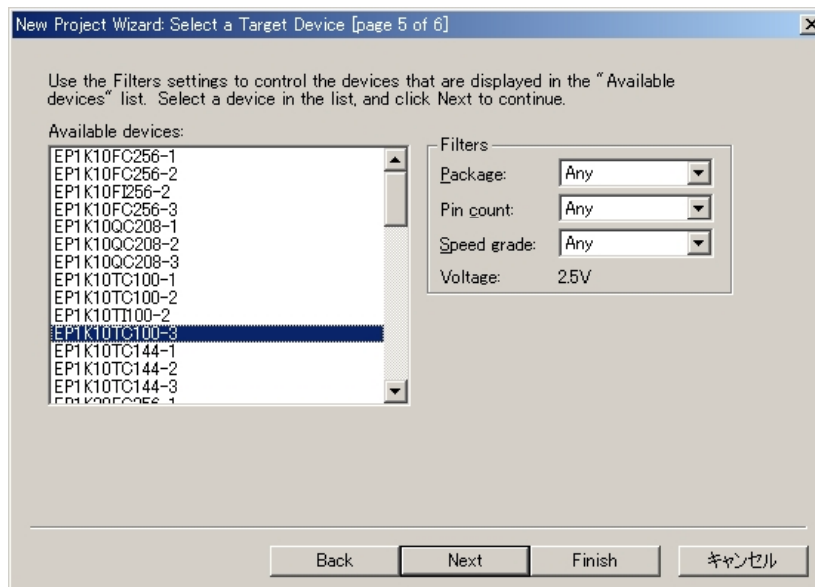
下の画面が、出ますが、何も入力せずに「Next」をクリックします。



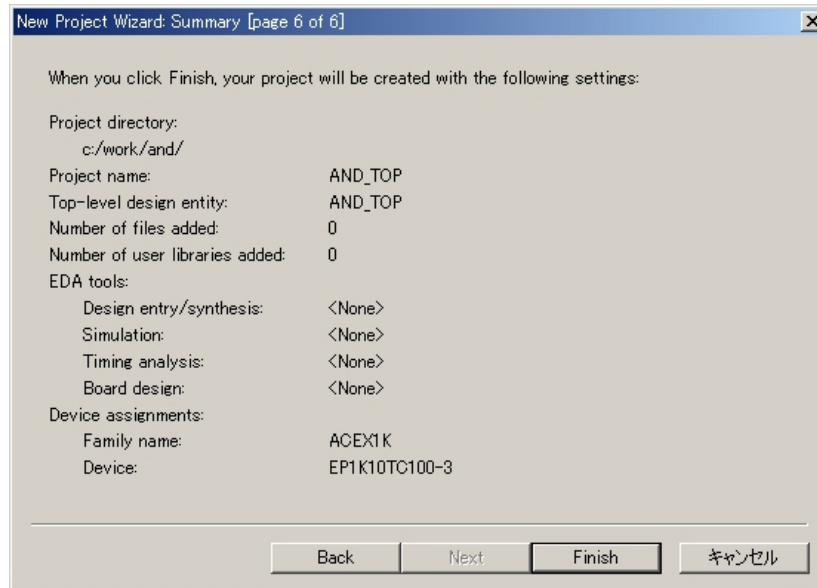
デバイスファミリーの指定画面が出ますので、ACEX1K を選択して「Next」をクリックします。



デバイスの指定画面が出ますので、EP1K10TC100-3 を選択して「Next」をクリックします。

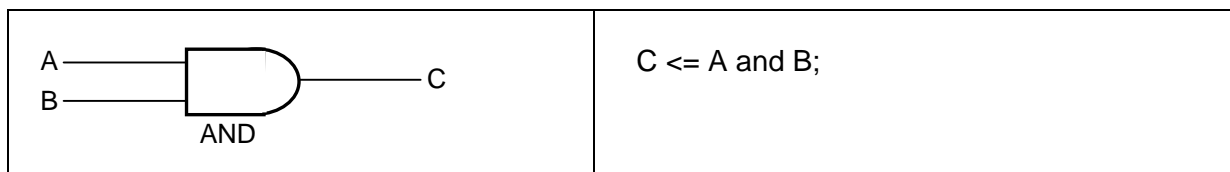


下の確認画面が、出ますので、間違いがなければ「Finish」をクリックしてください。



VHDL による AND 回路

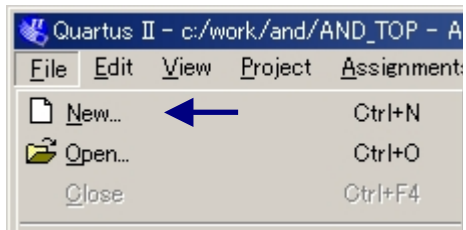
VHDL では、AND 回路は次のように記述します。



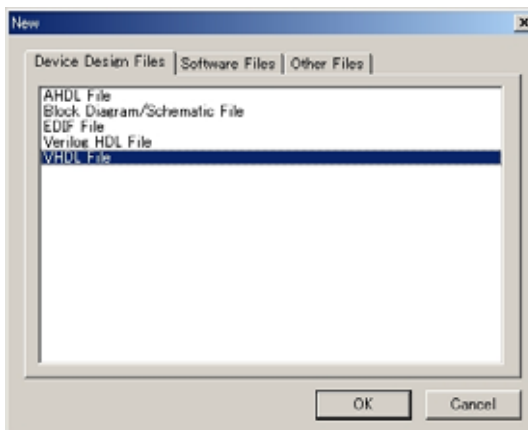
入力 A を「押しボタンスイッチ」の「SW3(A)」、入力 B を「押しボタンスイッチ」の「SW2(B)」とします。
出力 C は「赤色 LED」の「LED1」とします。実装する手順について説明します。

新規ソースの追加

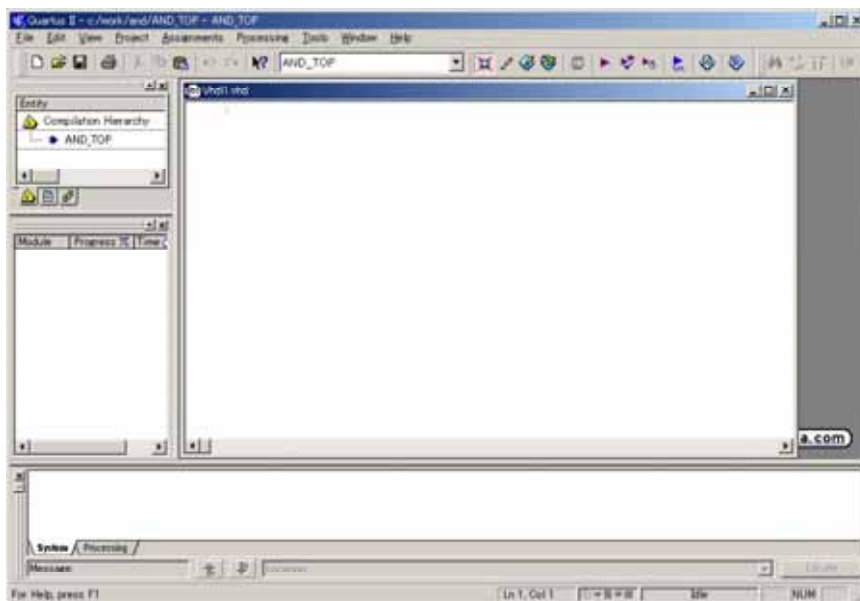
[File] - [New] を選択します。



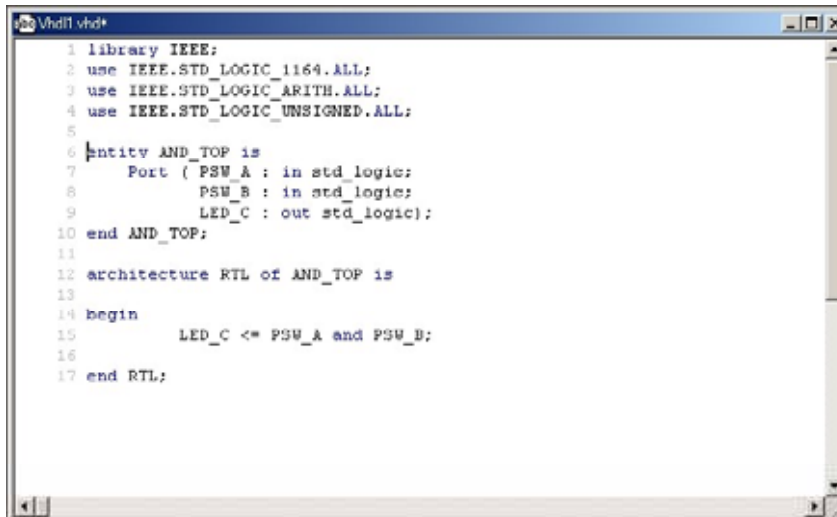
下の画面が、出ますので Device Design Files から VHDL File を選択して「OK」をクリックしてください。



下の画面のように白紙のテキストエディタが現れます。
(Quartus II では自動的に VHDL コードの雛型は、生成されません。)

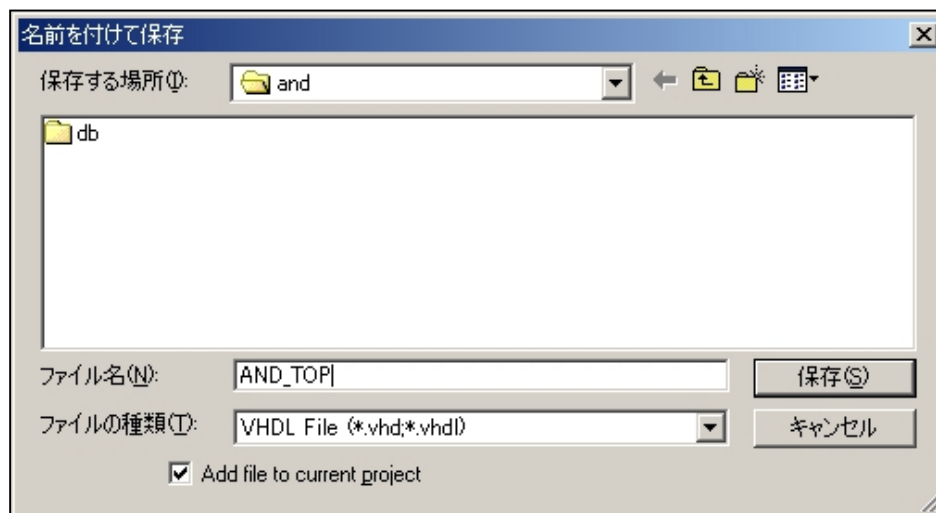


エディタに AND 回路の VHDL コードを入力します。



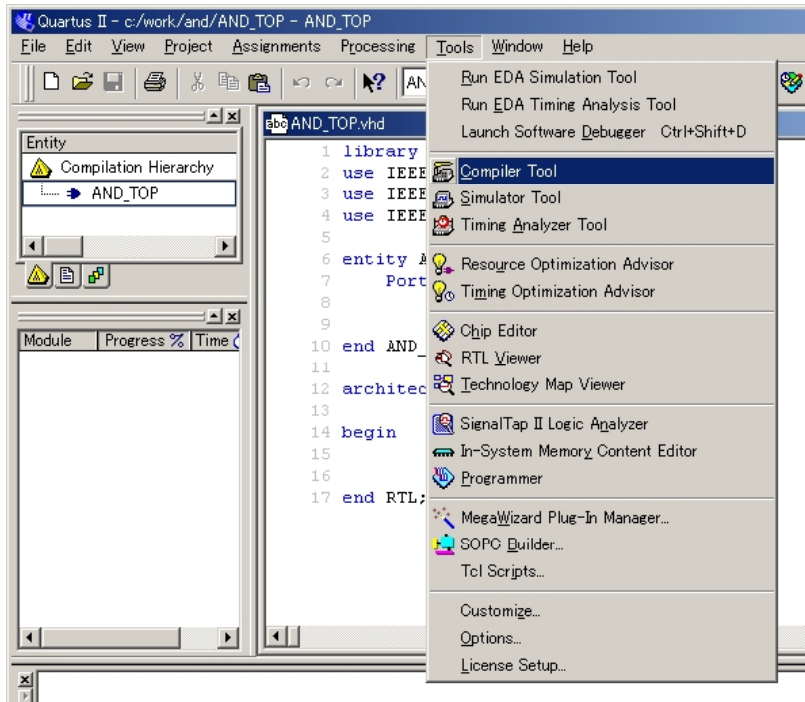
```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 use IEEE.STD_LOGIC_ARITH.ALL;
4 use IEEE.STD_LOGIC_UNSIGNED.ALL;
5
6 entity AND_TOP is
7     Port ( PSW_A : in std_logic;
8           PSW_B : in std_logic;
9           LED_C : out std_logic);
10 end AND_TOP;
11
12 architecture RTL of AND_TOP is
13
14 begin
15     LED_C <= PSW_A and PSW_B;
16
17 end RTL;
```

AND 回路の VHDL コードの入力が、終了したらファイル名を付けて保存します。

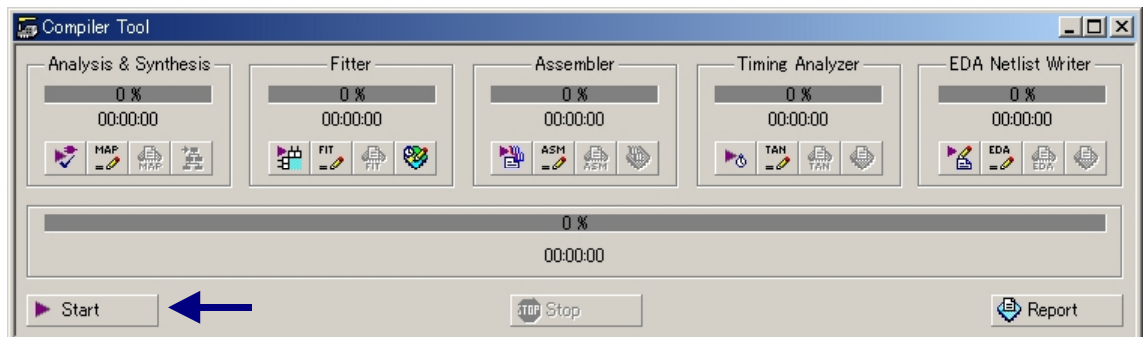


コンパイル

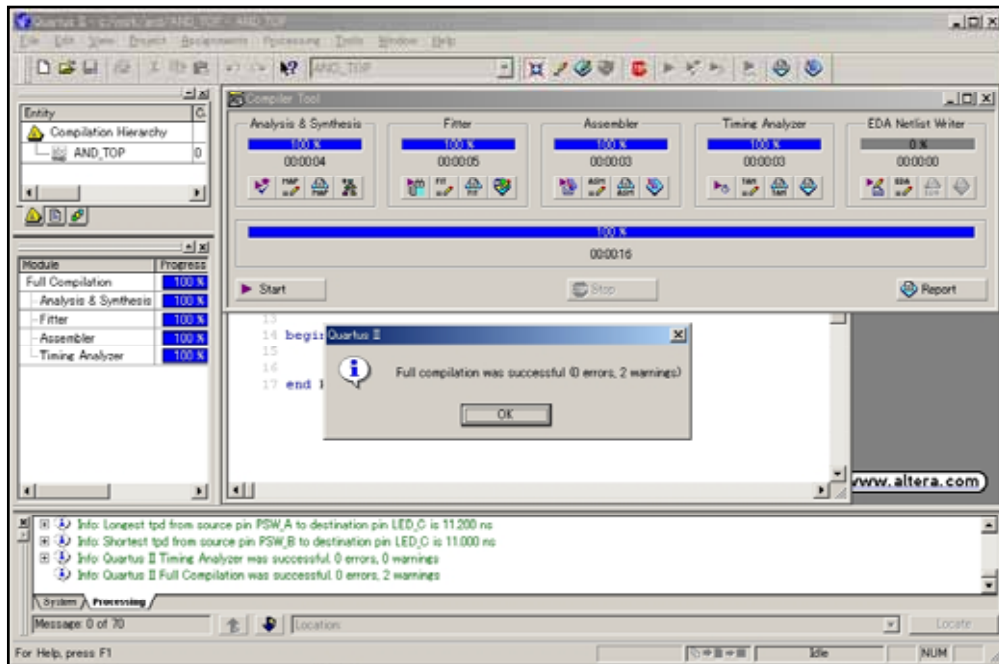
次に作成したソースコードのコンパイルを行います。
[Tools] - [Compiler Tool] を選択します。



下のような画面が、表示されますので、「Start」を押します。

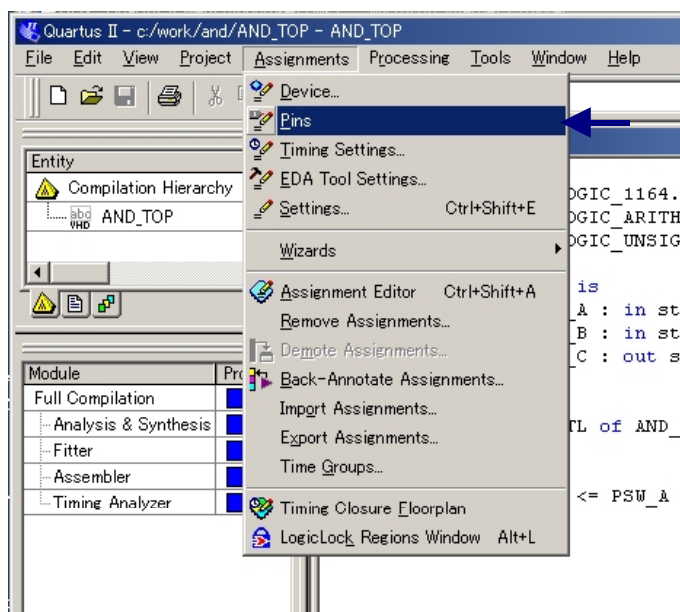


コンパイル処理が始まると処理状況が、表示されます。
 コンパイル処理が正常に終了すると下のような画面が、表示されますので「OK」をクリックしてください。

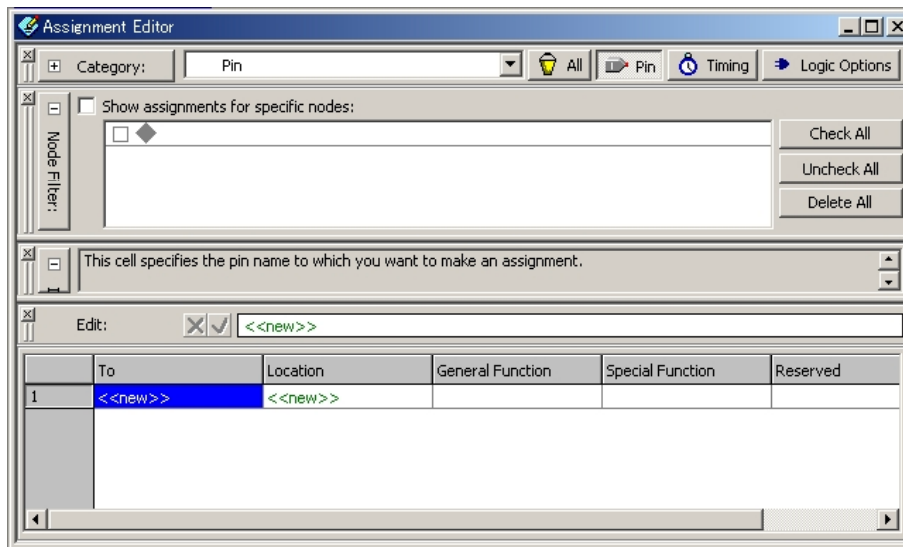


ピンの割り当て

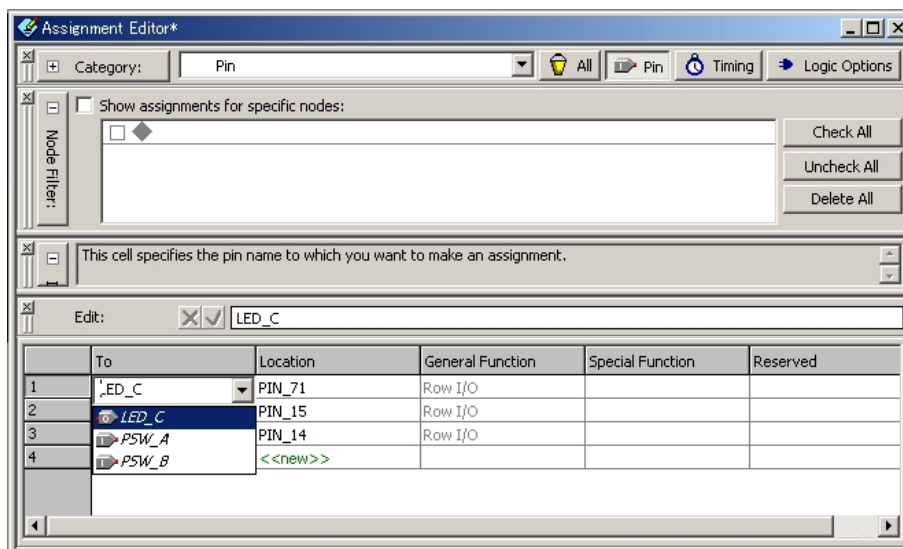
VHDL ソースファイルで宣言した入出力ピンを、実際のデバイスの何番ピンに割り振るかを指定します。[Assignments] - [Pins]を選択します。



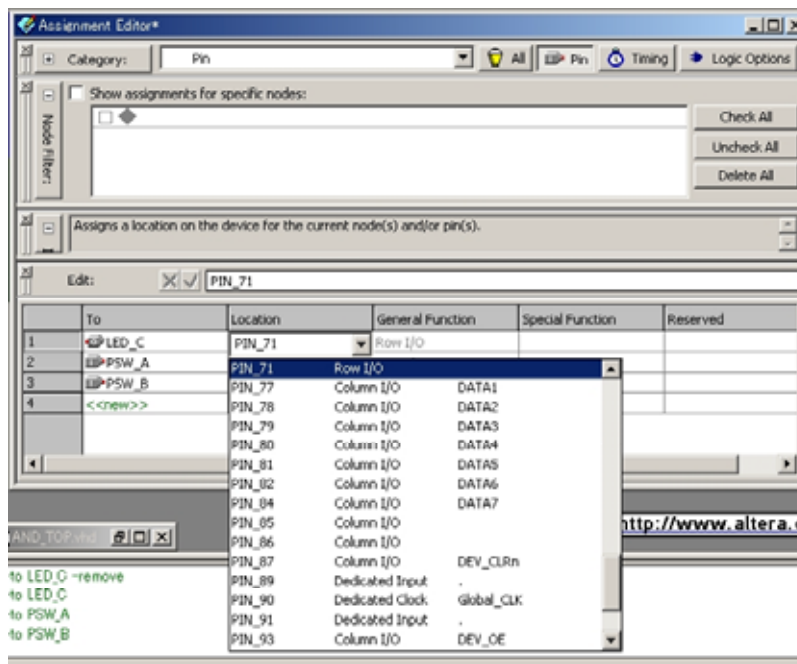
下のような画面が、です。ここで、VHDL のソースファイルで宣言した端子をデバイスのピンに割り当てます。



To の<<New>>をクリックすると VHDL のソースファイルで宣言した端子が、表示されるので選択します。

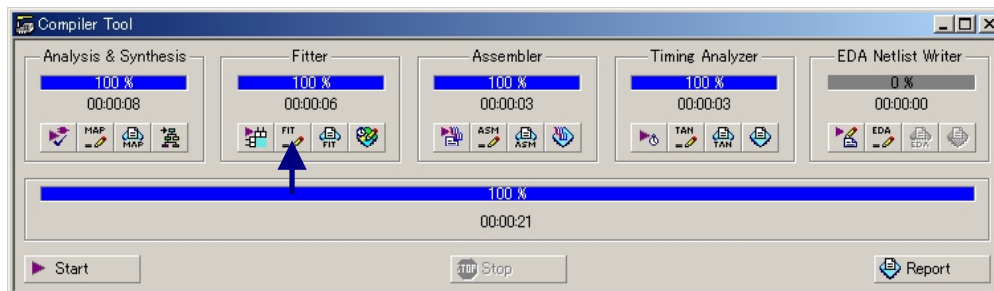


Location の<<New>>をクリックするとデバイスの端子番号が、表示されるので割り当てる端子番号を指定します。ピンの割り当てが、できたら保存します。

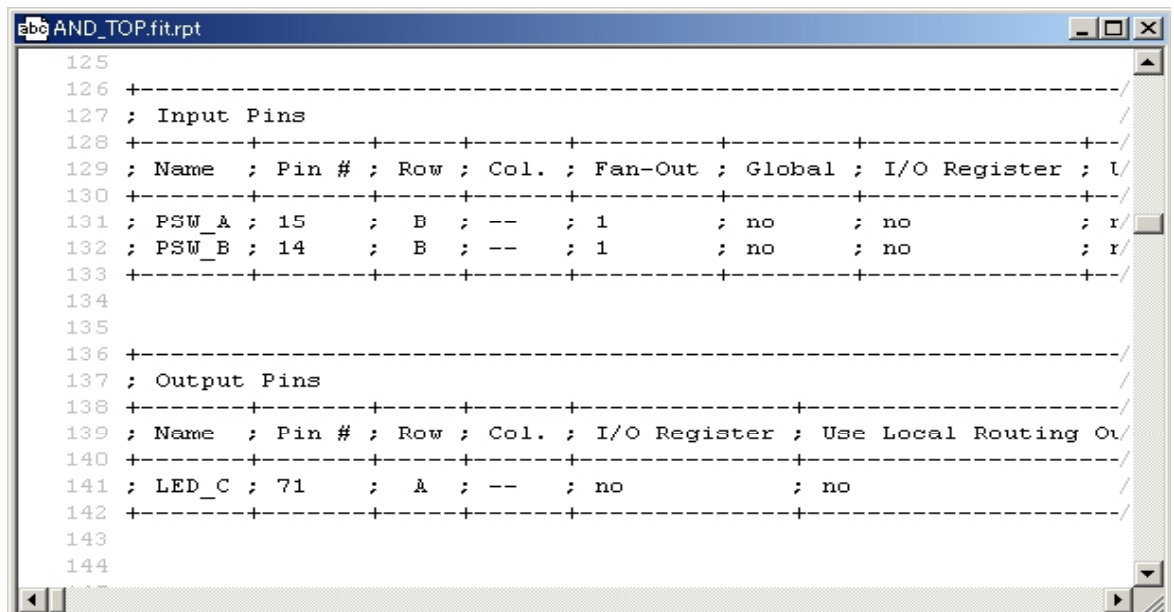


ピンの割り振りが、終了して保存したらコンパイルの手順で再度コンパイルを実行して Fitting Report を確認します。

Fitting Report は、矢印のボタンを押すと表示されます。

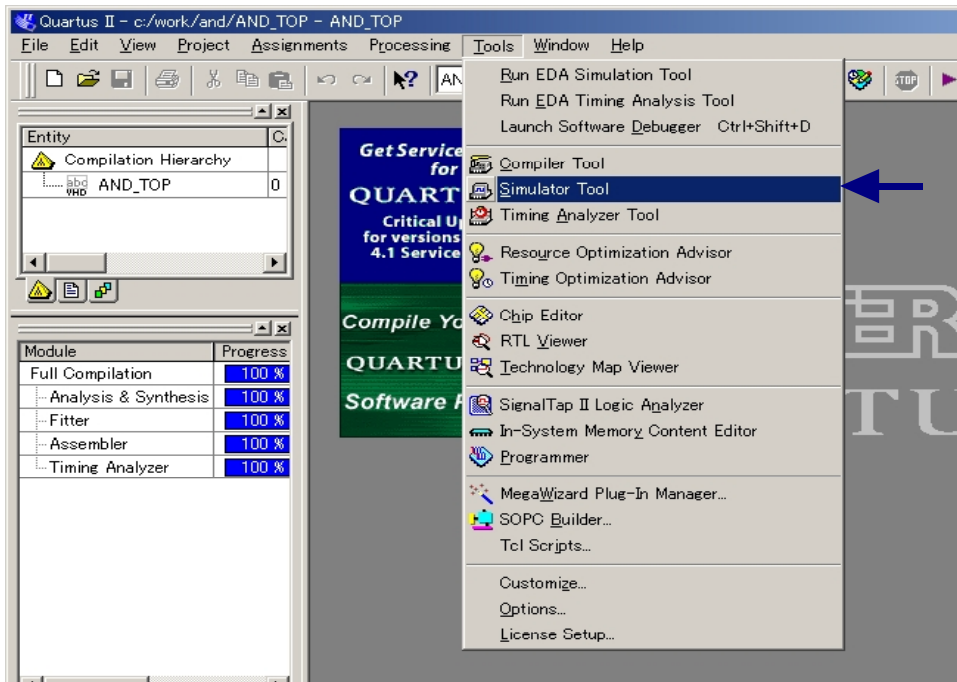


指定したピン番号になっていることを確認できます。

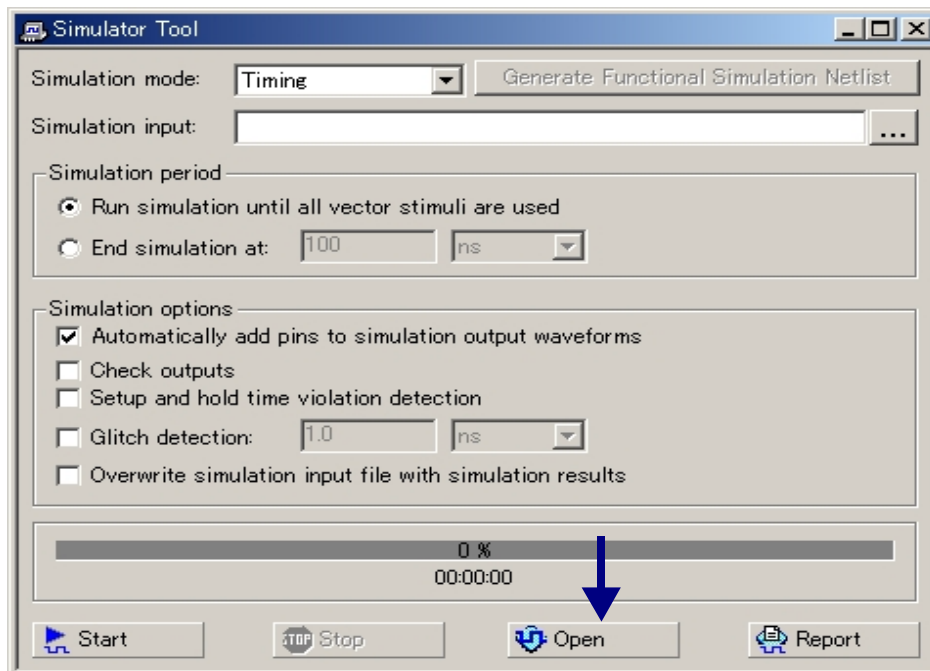


シミュレーション

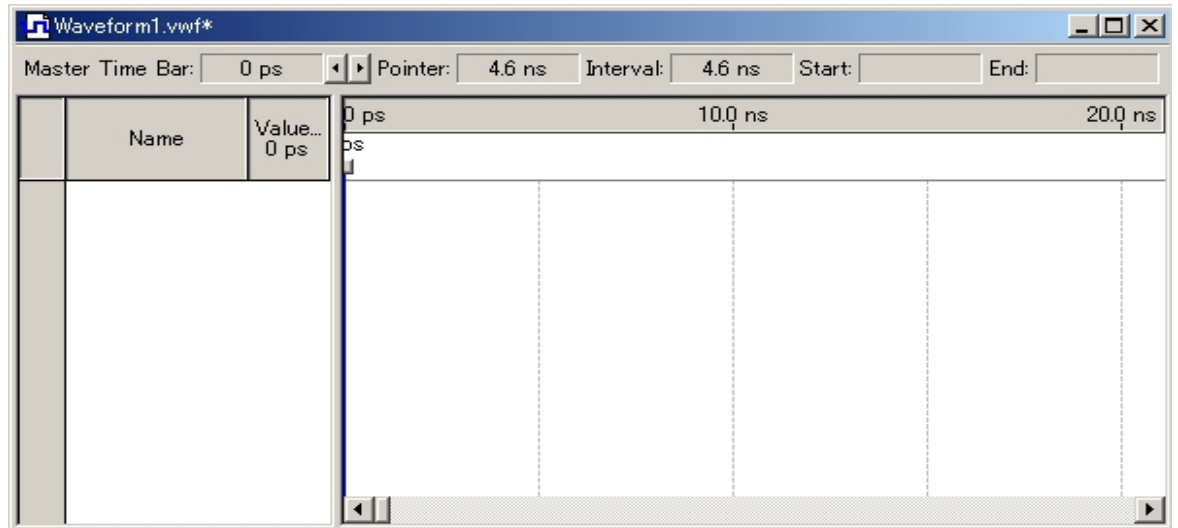
入出力のピンの割り当てに間違いがないことを確認できましたので、シミュレーションを行います。
[Tools] - [Simulator Tool] を選択します。



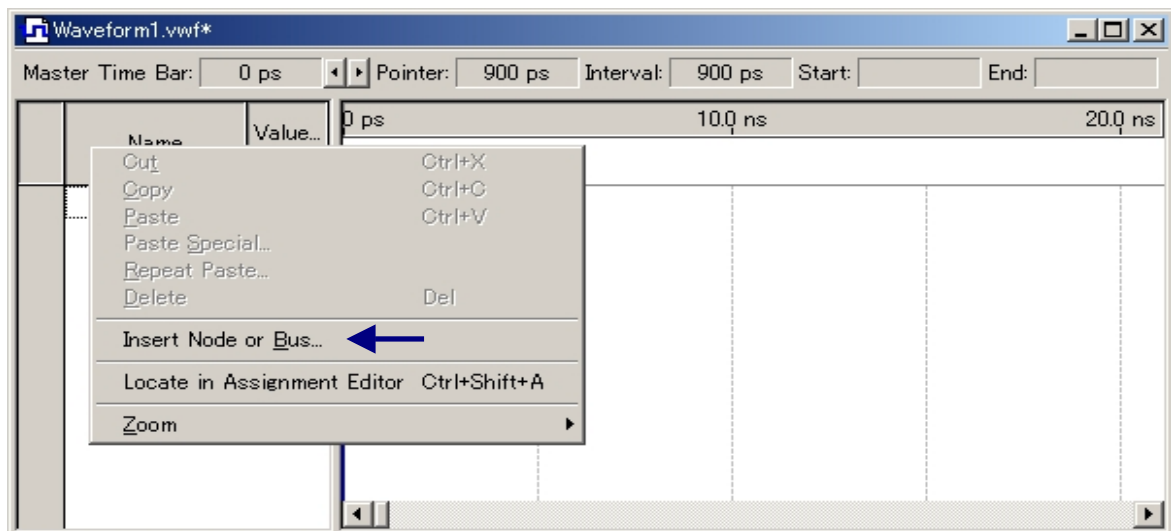
下のような画面が、できるので、画面の設定になっていることを確認して「Open」のボタンを押します。



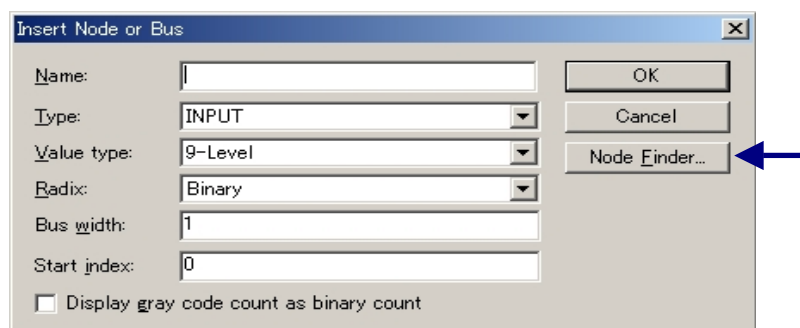
下のような画面が、できます。



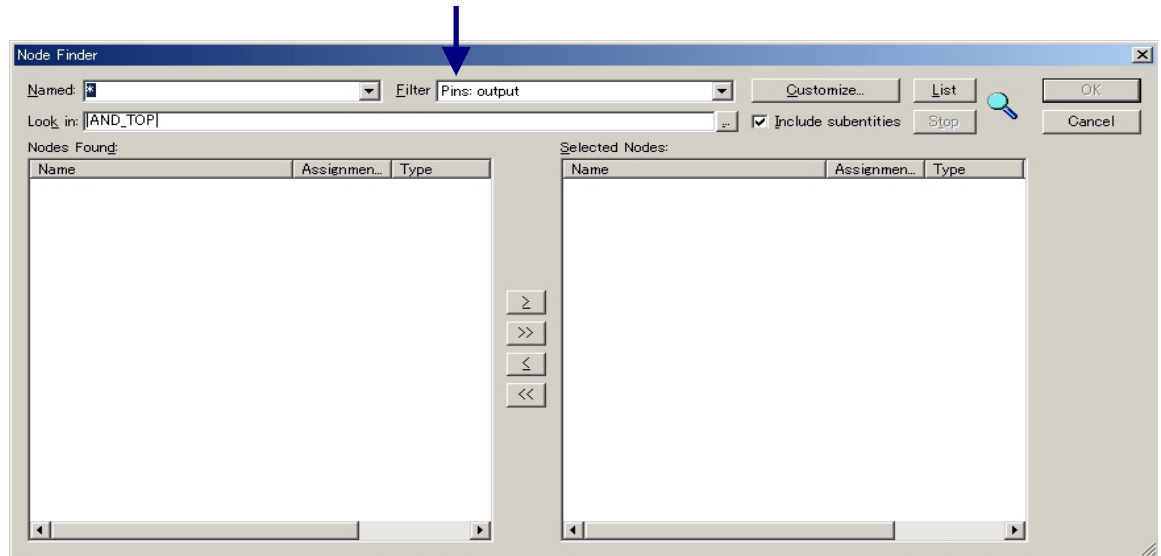
「Name」の部分を右クリックするとポップアップメニューが、開きます。



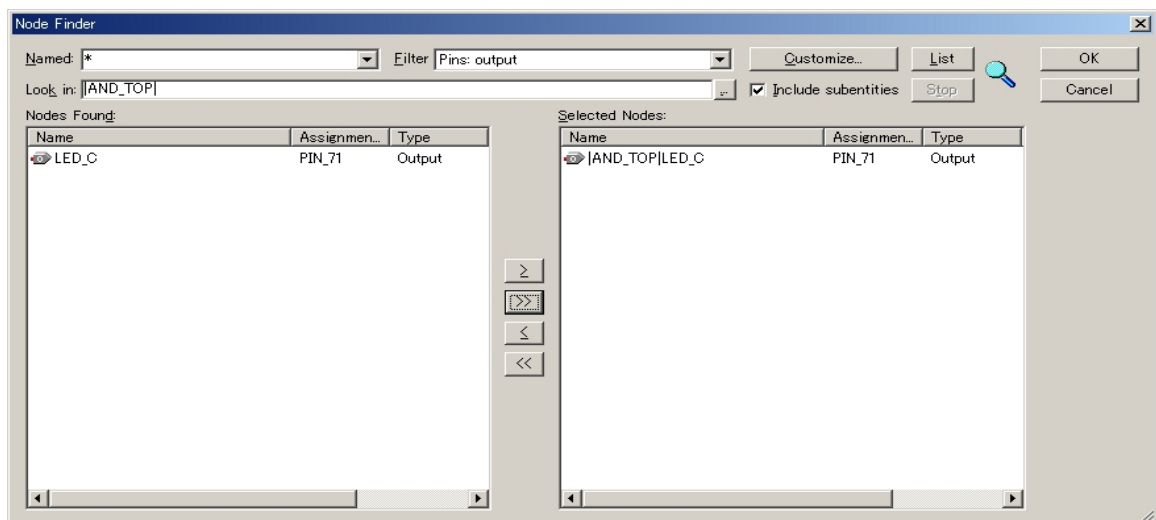
「Insert Node or Bus」を選択すると下の画面が、できるので、「Node Finder」をクリックします。



下の Node Finder の画面が、であるので「Filter」で「Pins: output」を選択して「List」をクリックすると「Nodes Found」に entity に宣言されている出力端子が、表示されます。

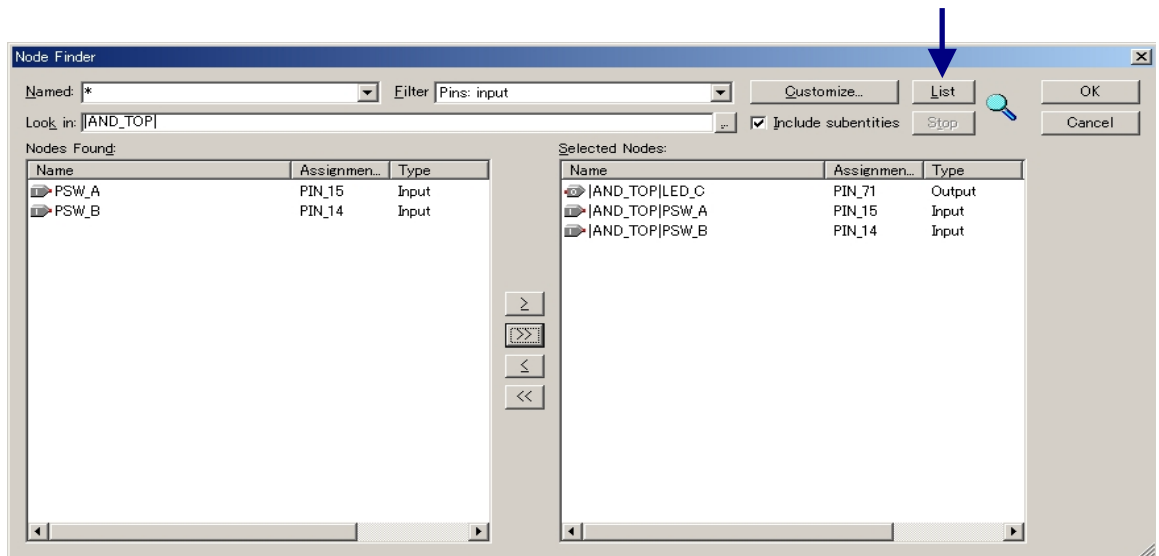


「Nodes Found」の「LED_C」を選択して画面中央の「>>」をクリックすると「Selected Nodes」に「LED_C」が、登録されます。

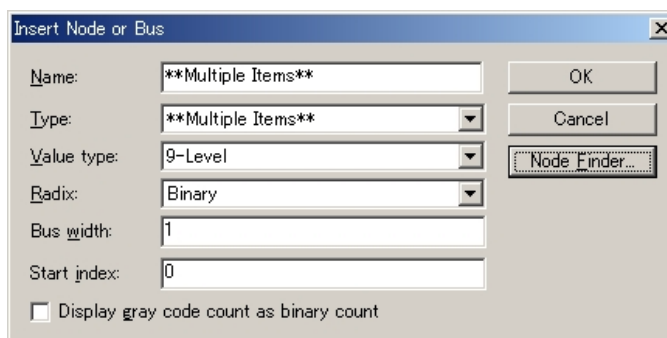


次に「Filter」で「Pins: input」を選択して「List」をクリックすると「Nodes Found」に entity に宣言されている入力端子が、表示されます。

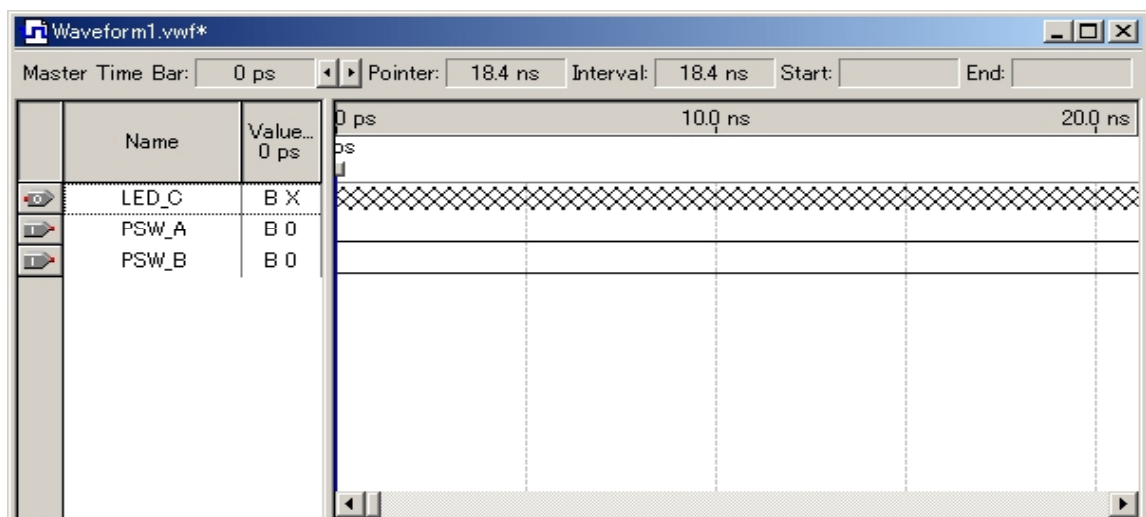
「Nodes Found」の「PSW_A」と「PSW_B」を選択して画面中央の「>>>」をクリックすると「Selected Nodes」に「PSW_A」と「PSW_B」が、登録されます。
必要なピンが、「Selected Nodes」の登録されたら「OK」をクリックします。



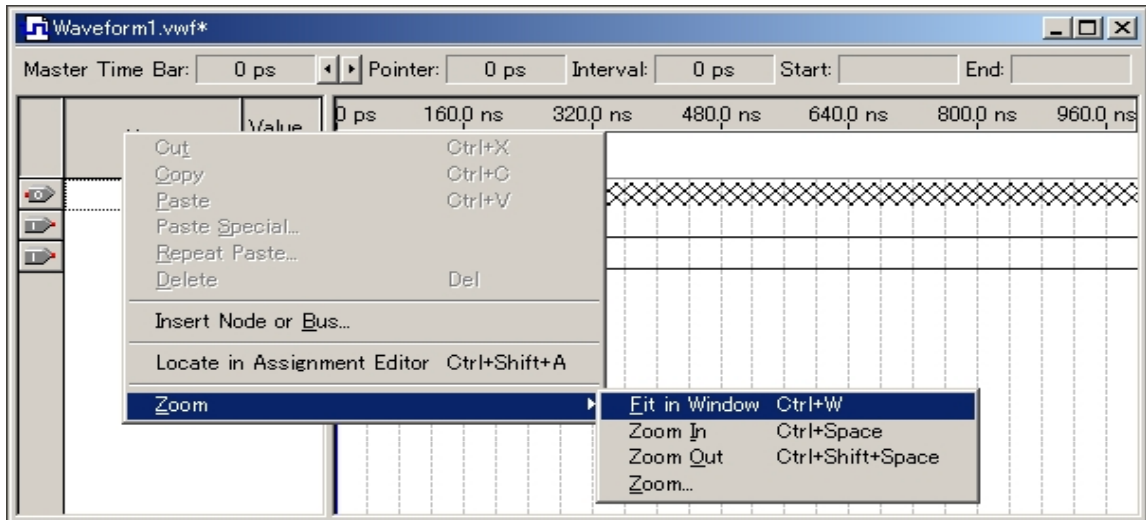
「Insert Node or Bus」の「Name」と「Type」の欄に「* * Multiple Items * *」と表示されていることを確認して「OK」をクリックしてください。



下の画面のようにシミュレーションを行うピンが、登録されます。

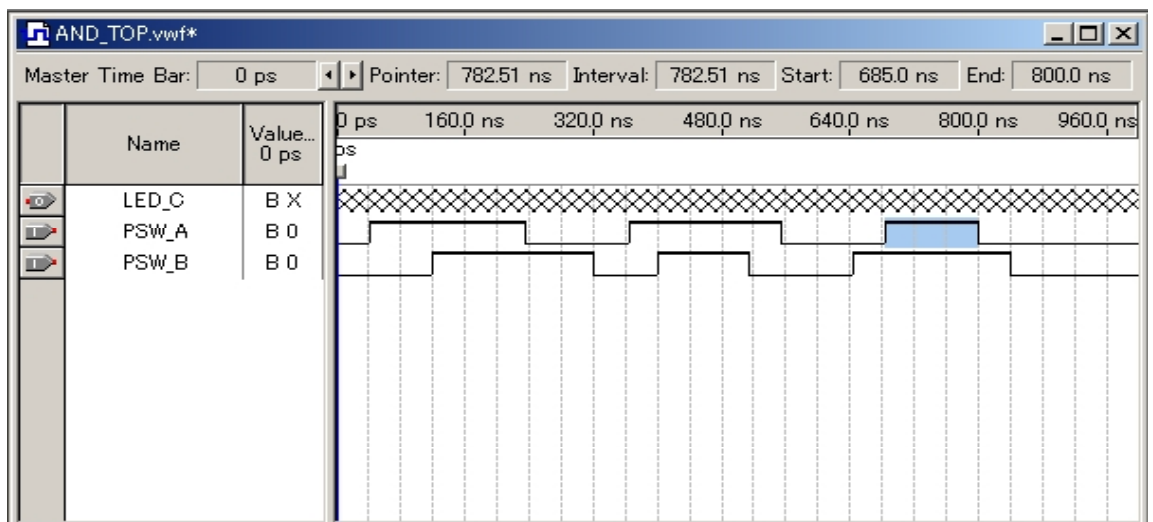
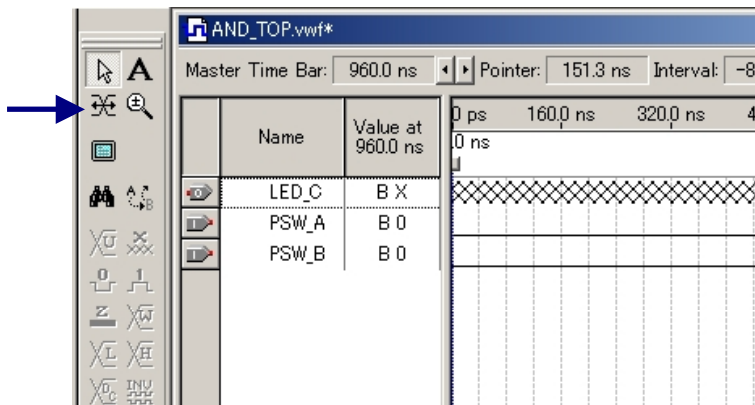


次に「Name」の部分をクリックして「Zoom」「Fit in Window」を選択すると画面右の時間軸が表示領域に合わせて表示されます。

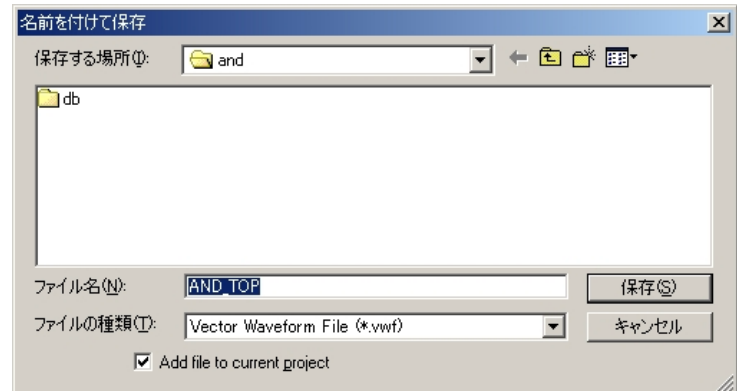


シミュレーションに必要な波形を「PSW_A」と「PSW_B」に入力します。

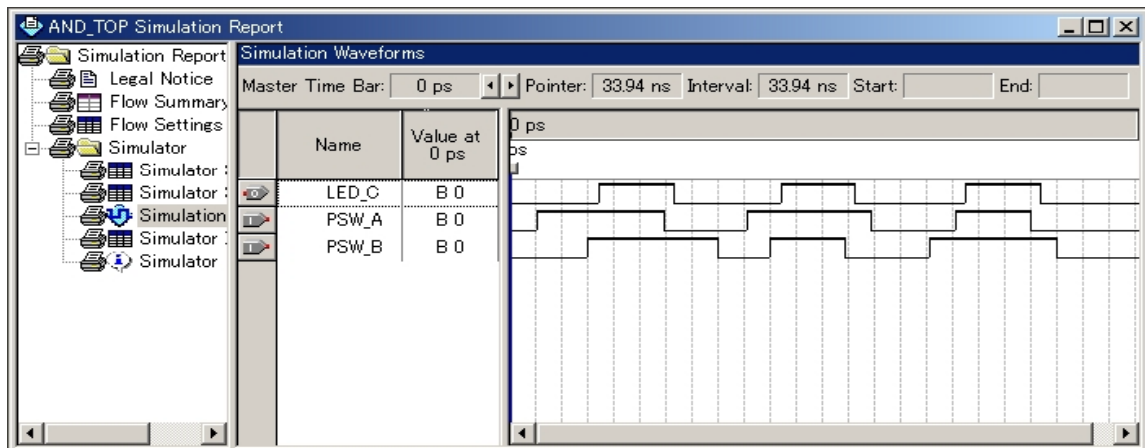
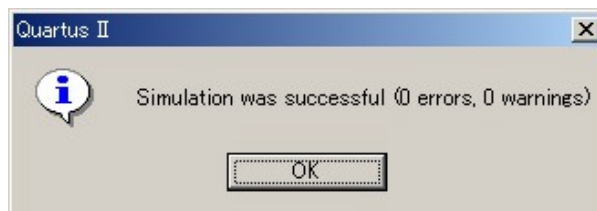
シミュレーションに必要な波形の入力は、赤い矢印のマークをクリックして「PSW_A」と「PSW_B」の波形表示エリアの適当な部分をマウスで、ドラッグして動かすとドラッグした範囲にパルス波形が現れます。



シミュレーションを行う波形を「PSW_A」と「PSW_B」に設定できたら「Simulator Tool」の画面に戻って「Start」ボタンを押すと vwf ファイルを保存する確認画面が、でますので、「はい」をクリックすると保存ファイル名を聞いてきますので、「ファイル名」を「AND_TOP」として「保存」をクリックするとシミュレーションが、行われます。



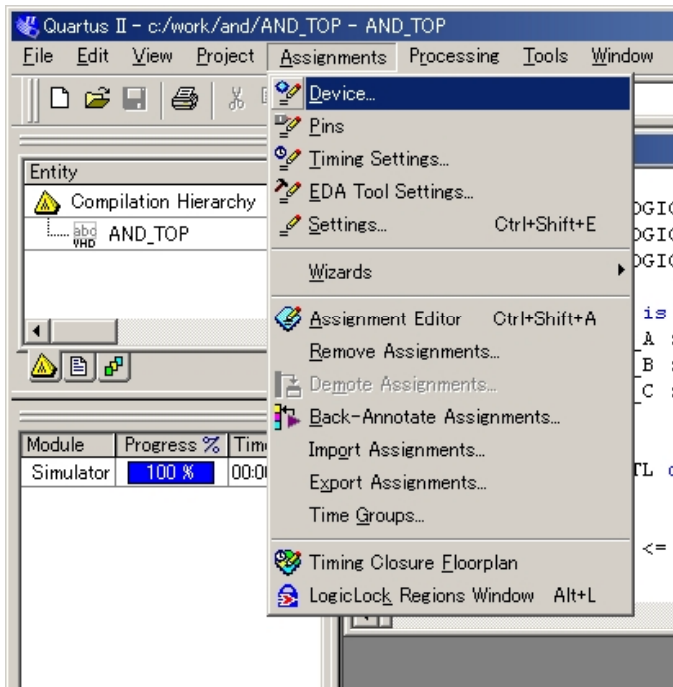
シミュレーションが、成功すると下の画面が、でますので、「OK」をクリックして「Simulator Tool」画面の「Report」ボタンを押すとシミュレーション結果が、表示されます。



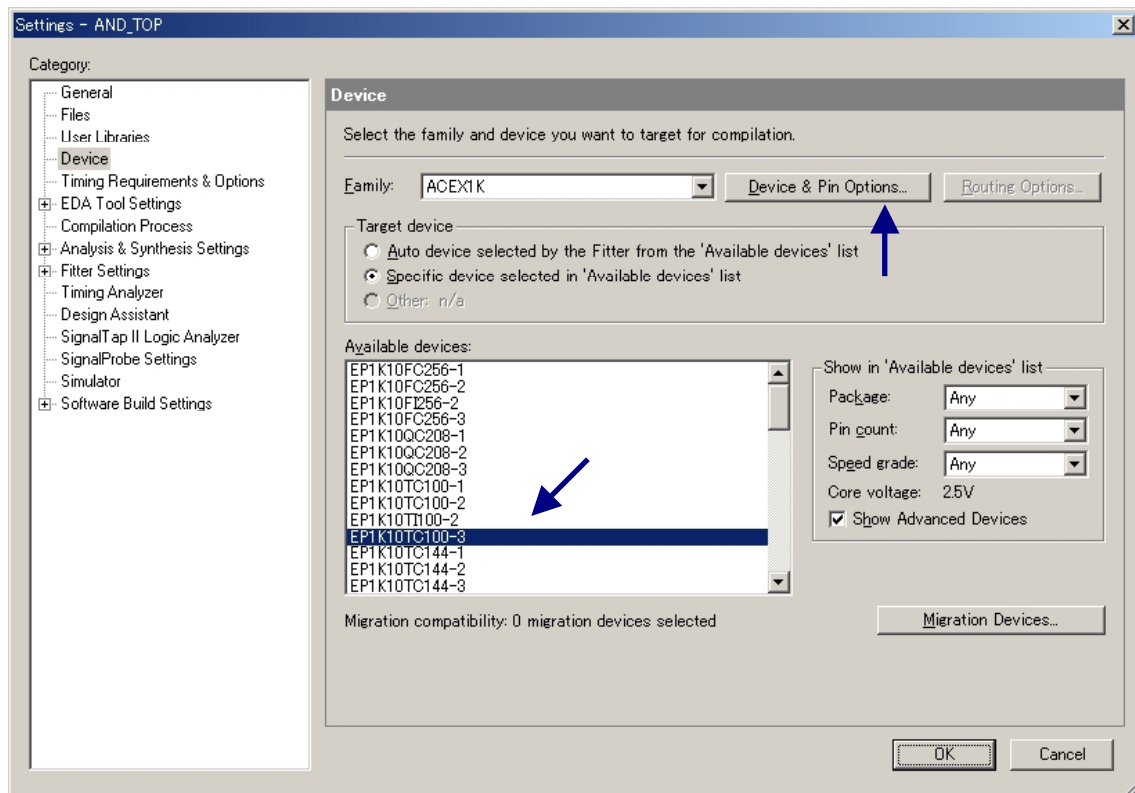
RBF ファイルの作成

RBF ファイルを生成するためには、使用しないピンをトライステートに設定してコンパイル時に
* .rbf ファイルを生成するように設定する必要があります。

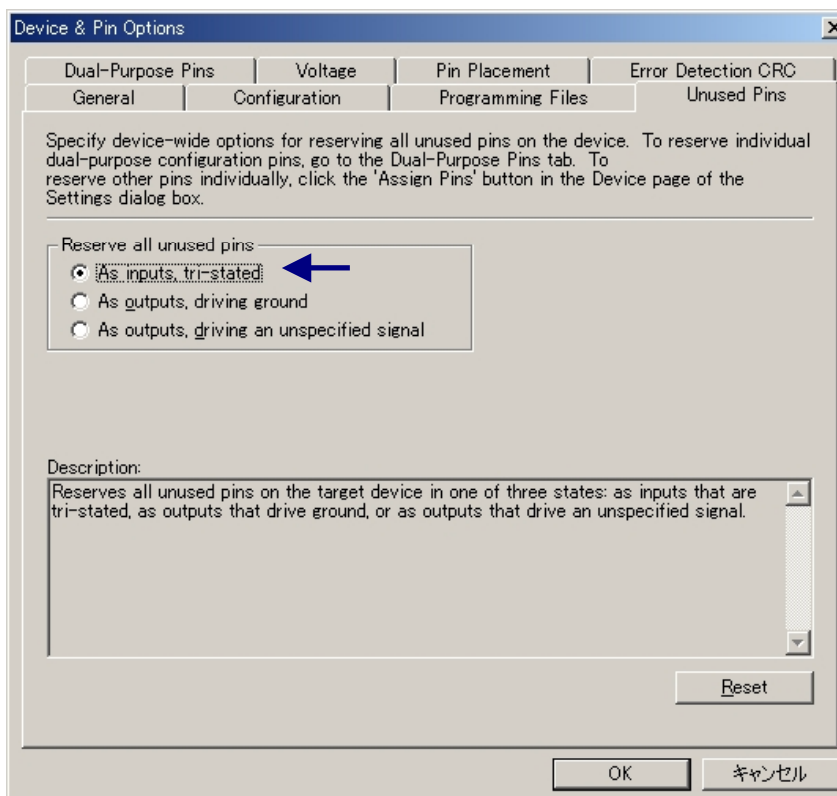
「Assignments」 - 「Device」を選択します。



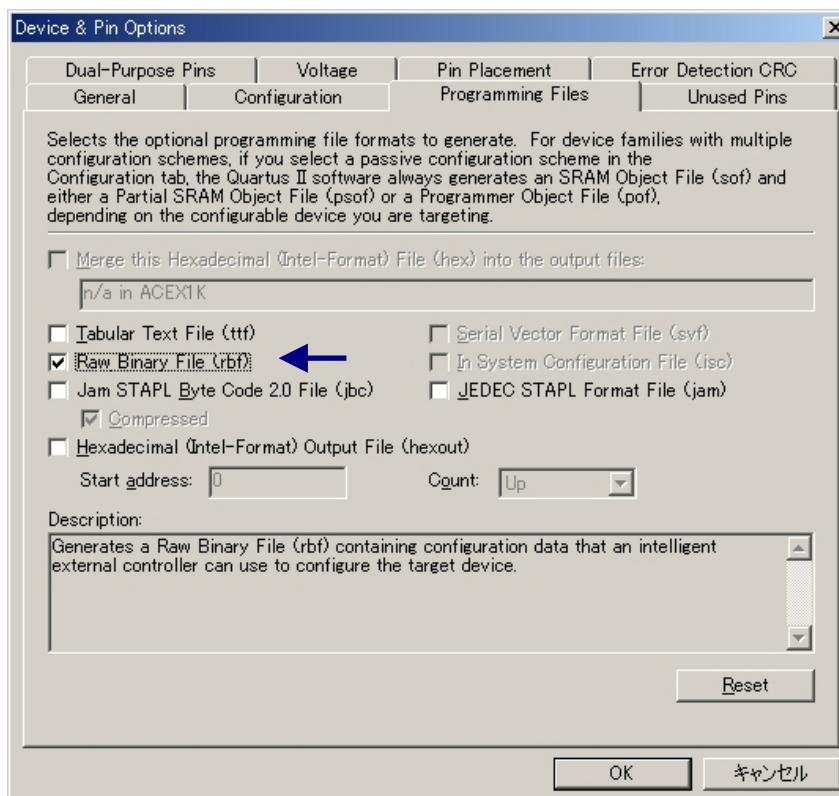
以下の画面が、表示されますので、デバイスが、EP1K10TC100-3であることを確認して「Device & Pin Options」をクリックしてください。



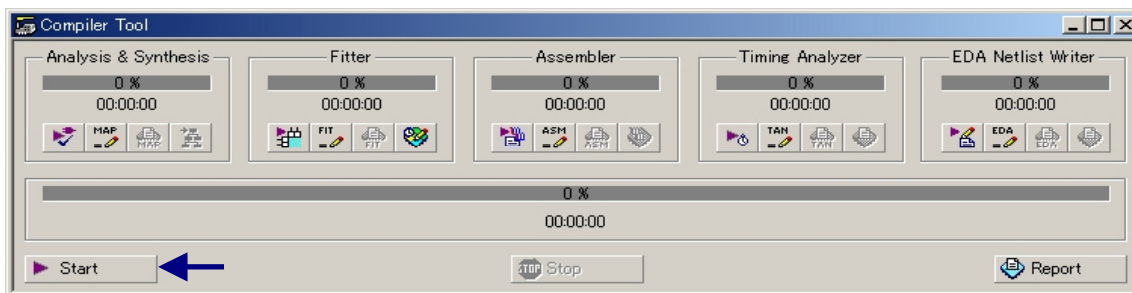
「Device & Pin Options」の設定画面が開きますので、「Unused Pins」の「Reserve all unused pins」の設定で、「As inputs, tri-stated」をチェックしてください。



次に「Programming Files」の設定画面を開き「Raw Binary File (rbf)」をチェックします。



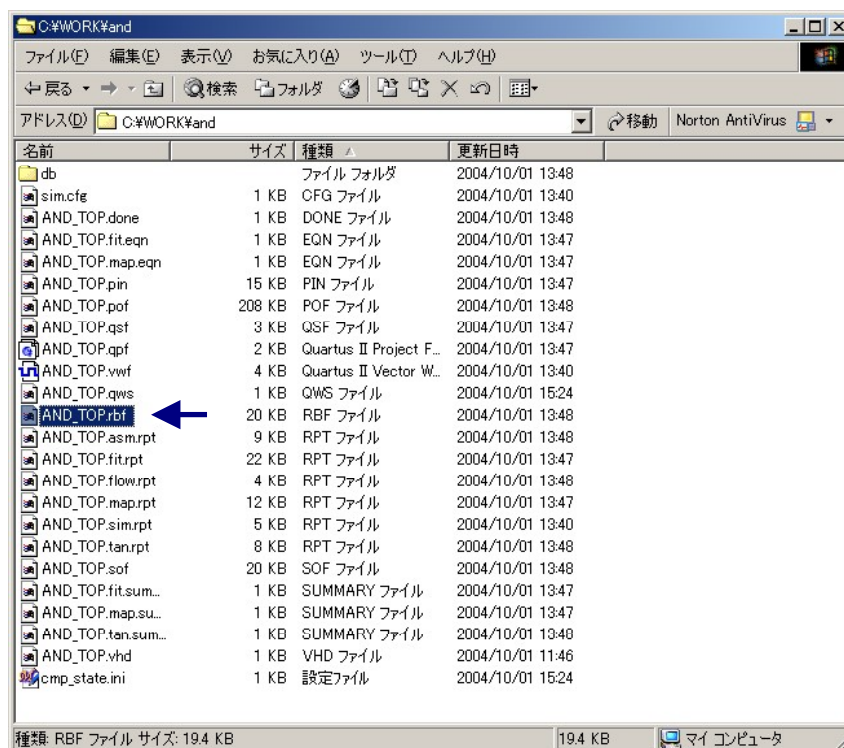
.rbf ファイルを生成するためにコンパイルと同じ手順で、[Tools] - [Compiler Tool] を選択し「Commpailer Tool」の「Start」ボタンをクリックしてください。



コンパイルが、成功すると下の画面がでますので、「OK」をクリックしてください。



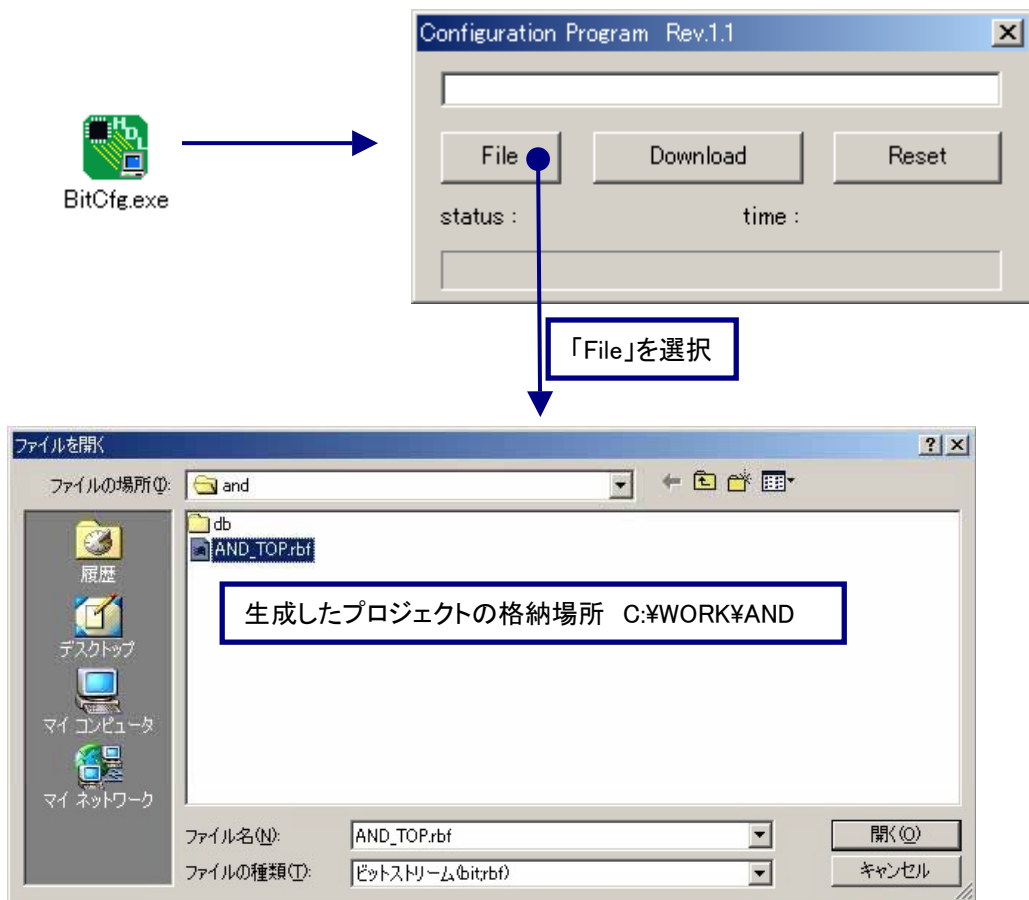
これで C:\¥WORK¥AND に AND_TOP.rbf が、生成されていることを確認してください。



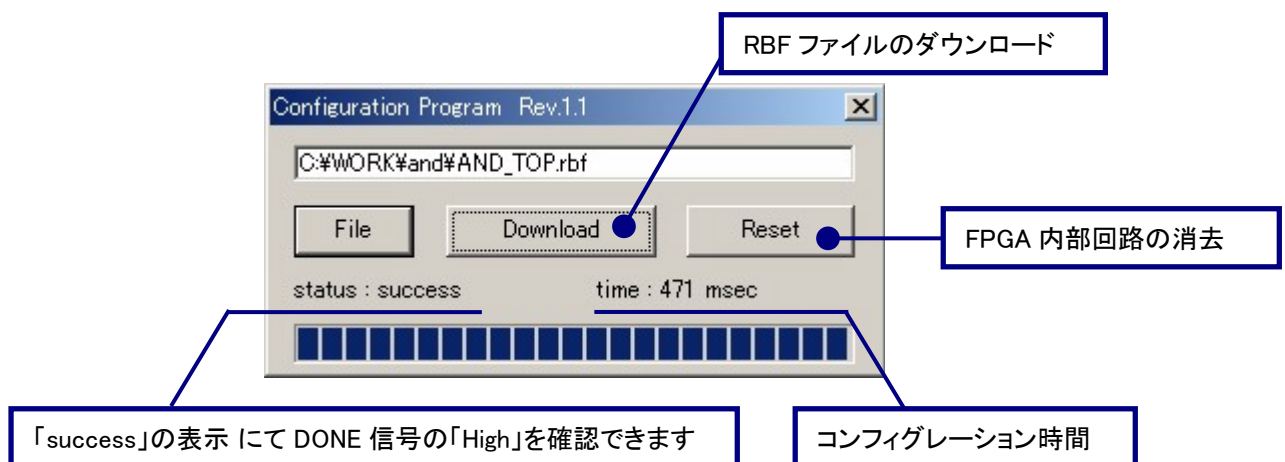
EDA-002 の FPGA へのコンフィグレーションは、USB 経由で行うため ALTERA 社のダウンロードツールを利用することはできません。付属の専用ソフトウェアにて USB コンフィグレーションをおこなってください。

専用ソフトウェアによる FPGA コンフィグレーション

付属 CD 内にある「BitCfg.exe」を起動させると、次のウィンドウが起動します。

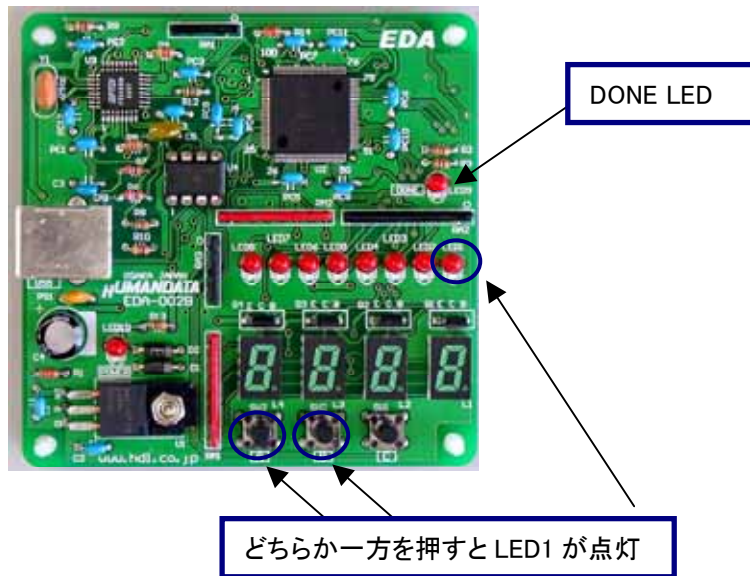


生成した AND 回路「and_top.rbf」を選択し、「Download」を選択します。「DONE LED」点灯し、ウィンドウの「status:success」であれば正常に FPGA にコンフィグレーションであることを意味します。

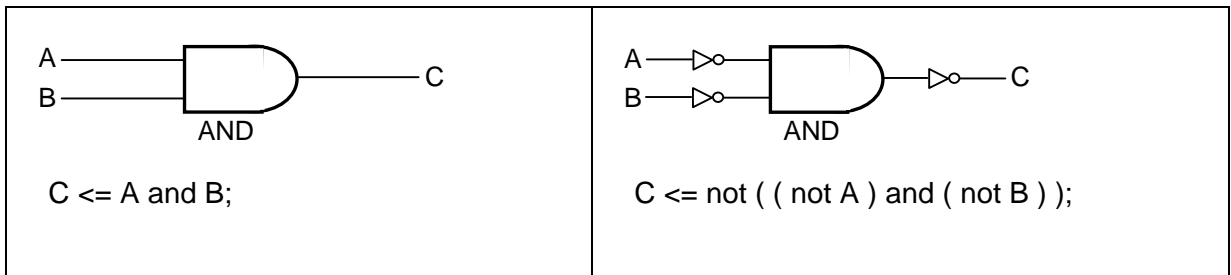


動作の確認

赤色 LED と押しボタンスイッチが負論理であるため、押すと点灯するのではなく両方の押しボタンを押していない状態で消灯し、それ以外は点灯となっています。



両方、押した時だけ点灯させるには、次のように「インバータ」を挿入します。

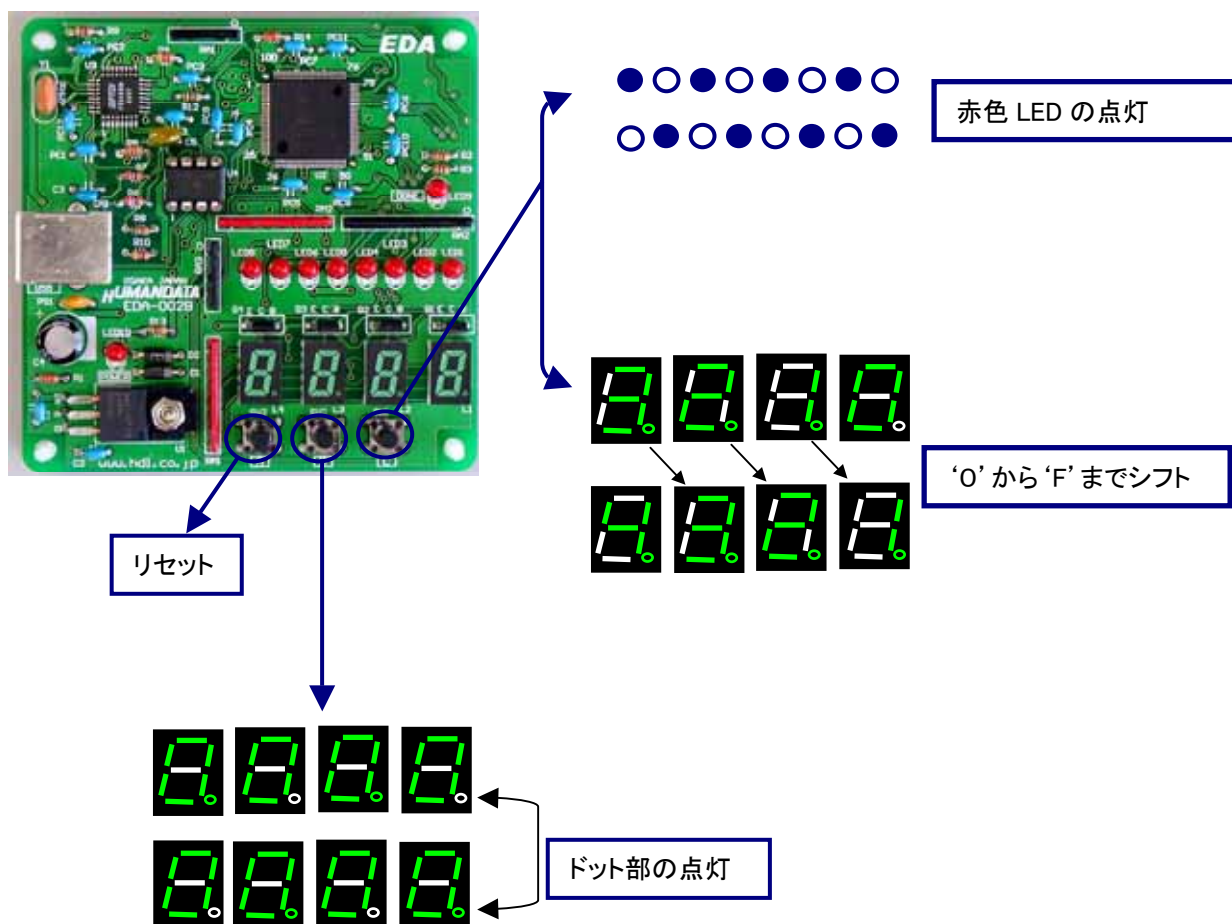


EDA-002 サンプル回路

これまでに説明した回路をもとに、サンプル回路の VHDL ソースコードを含む Quartus プロジェクトを付属の CD に収録しています。EDA-002 用の USB ドライバがインストールされた状態で FPGA のコンフィグレーションを行ってください。

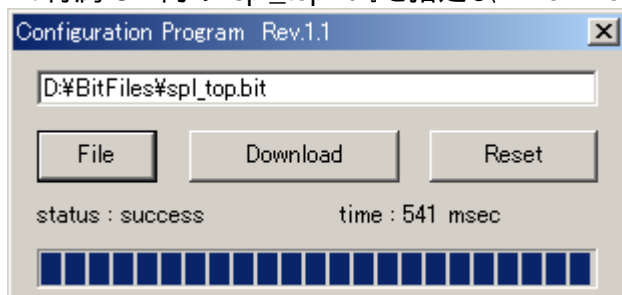
サンプル回路の動作

押しボタンスイッチ『A』『B』『C』を押すことによって、7セグメントLEDと赤色LEDが点灯します。以下に動作を図示します。



サンプル回路のコンフィグレーション手順

1. 付属 CD 内の『BitCfg.exe』を起動する
2. 付属 CD 内の『spl_top.rbf』を指定し、『Download』ボタンを押す。



付属 CD-ROM の内容

【BitCfg】

FPGA コンフィグレーションのためのソフトウェア
Visual C++ のソースコード

【DirectDriver】

USB ドライバ (FTDI 社ダイレクトドライバ)

【EDA-002】

【DataSheet】

EP1K10TC100-3 (ALTERA 社 FPGA) acex.pdf
FT245BM (FTDI 社 USB FIFO) ds245b14.pdf

【Quartus】

「AND 回路 (and.zip)」と「応用回路 (spl.zip)」の Quartus II プロジェクト

【Document】

EDA-002 ユーザーマニュアル (カラー) EDA002man.pdf
EDA-002 回路図 EDA002B.sch.pdf

【RbfFiles】

マニュアルで説明した「AND 回路 (and.rbf)」と「応用回路 (spl.rbf)」の rbf ファイル

【EDX-002】

【DataSheet】

XC2S15VQ100 (XILINX 社 FPGA) ds001.pdf
FT245BM (FTDI 社 USB FIFO) ds245b14.pdf

【ISEprj】

「AND 回路 (and.zip)」と「応用回路 (spl.zip)」の ISE プロジェクト

【Document】

EDX-002 ユーザーマニュアル (カラー) EDX002man.pdf
EDX-002 回路図 EDX002C.sch.pdf
ModelSim インストールマニュアル modelsim.html

【BitFiles】

マニュアルで説明した「AND 回路 (and.bit)」と「応用回路 (spl.bit)」の BIT ファイル

最新のデータシートは、各社のホームページからダウンロードしてください。

XILINX 社のホームページ <http://www.xilinx.com/>

ALTERA 社のホームページ <http://www.altera.com/>

FTDI 社のホームページ <http://www.ftdichip.com/>

ACEX FPGA トレーナ
EDA-002

ユーザーズマニュアル

2004/10/05 第2版

有限会社ヒューマンデータ

〒567-0034

大阪府茨木市中穂積1-2-51

シャトー春日第3ビル5F

TEL 072-620-2002

FAX 072-620-2003

URL <http://www.hdl.co.jp>

Mail support@hdl.co.jp
