MAX II/Cyclone II

EP2C8 ボード

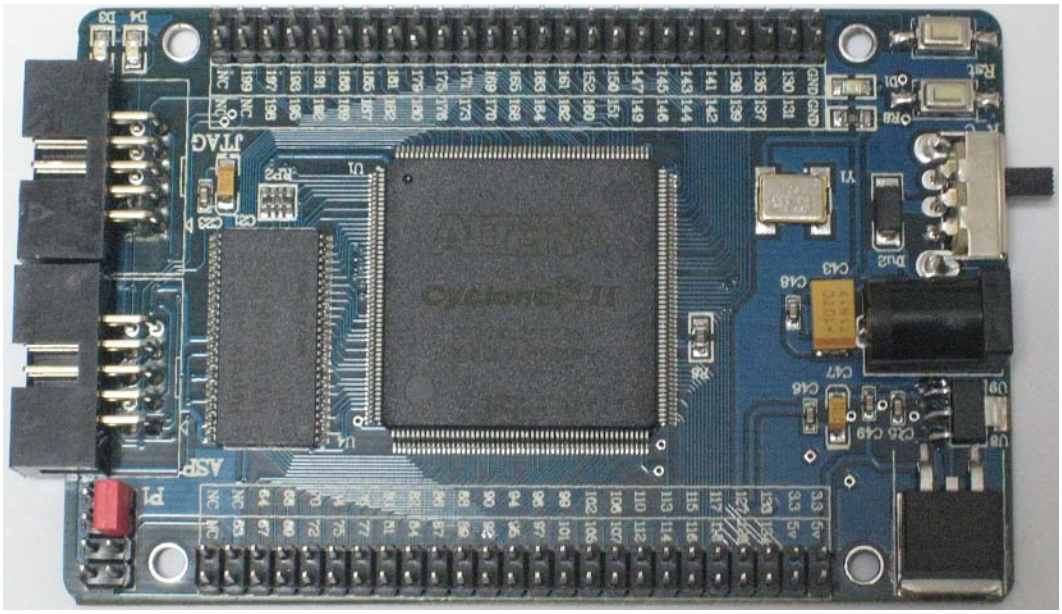
マニュアル

株式会社日昇テクノロジー

http://www.csun.co.jp

info@csun.co.jp

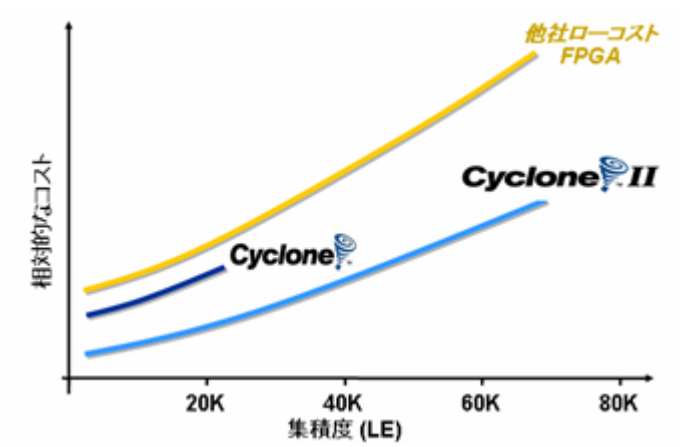
2010/05/27



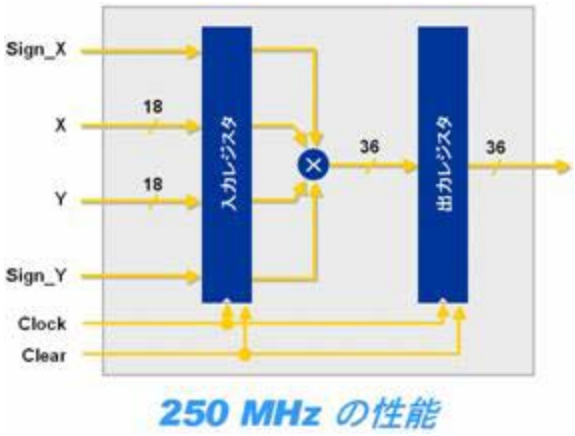
※ 使用されたソースコードはhttp://www.csun.co.jp/からダウンロードできます。

第一章 Cyclone II/EP2C8Q208 ボードの概要

Cyclone II デバイスは、90-nm テクノロジの優位性（小型ダイ・サイズ、高集積度、および低コスト）と、低コスト FPGA における最速性能を提供します。すべての Cyclone II デバイスは、TSMC の 90-nm プロセス技術と low-k 低誘電材を使用して 300-mm ウェハ上に製造されています。

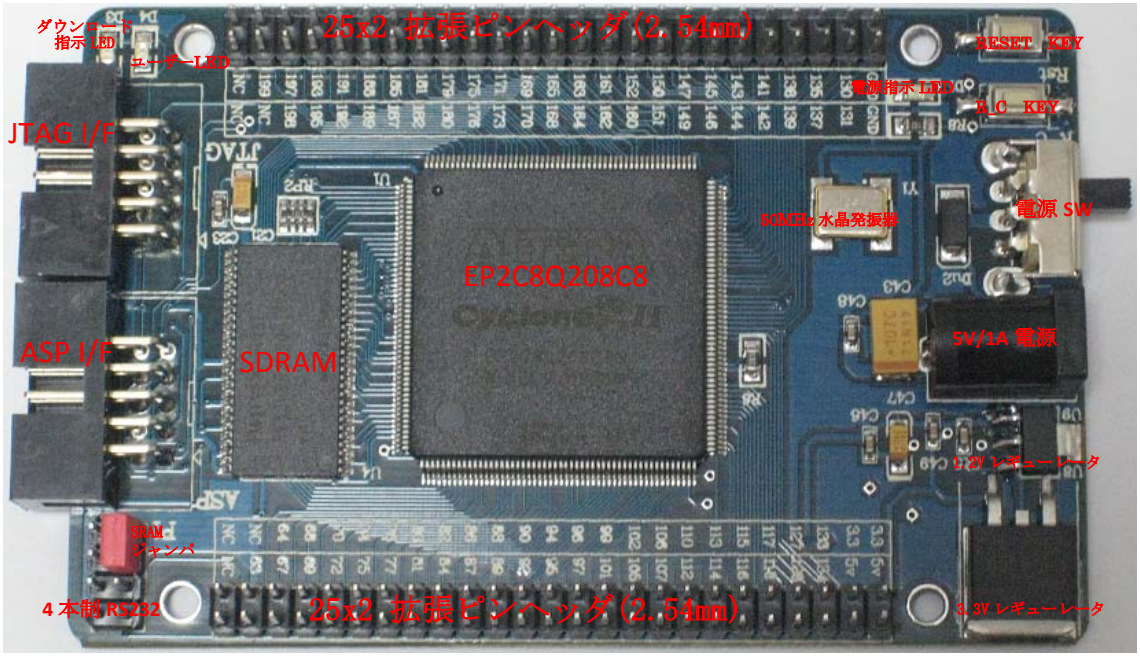


Cyclone II デバイスは、一般的なデジタル信号処理（DSP）機能を実装できる、最大 150 個の 18 ビット x 18 ビット・マルチプライヤを備えています。エンベデッド・マルチプライヤは、ロジック・エレメント（LE）ベースのマルチプライヤと比較してより高い性能とロジック効率を提供します。

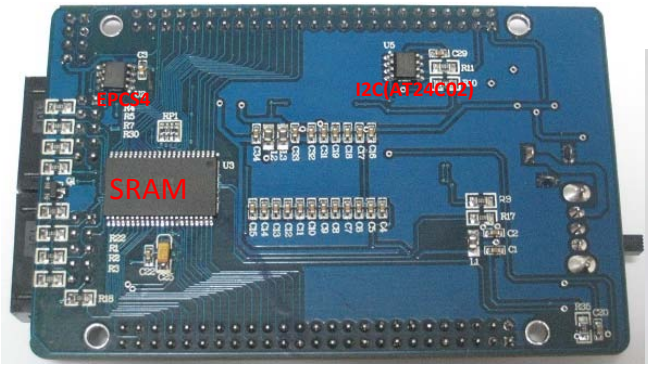


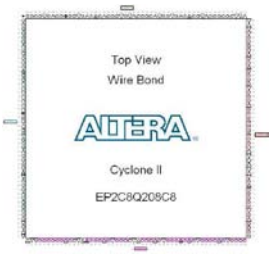
	デバイス					
特徴	EP2C5	EP2C8	EP2C20	EP2C35	EP2C50	EP2C70
ロジック・エレメント数	4,608	8,256	18,752	33,216	50,528	68,416
M4K RAM ブロック数	26	36	52	105	129	250
RAM 総ビット数	119,808	165,888	239,616	483,840	594,432	1,152,000
エンベデッド乗計算器数	13	18	26	35	86	150
PLL 数	2	2	4	4	4	4

1.1 概要仕様：

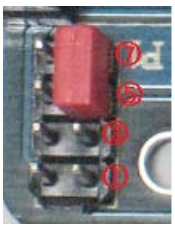


裏面と側面のイメージ：





- EP2C8Q208C8 FPGA マイコン搭載
- NIOS II ソフトプロセッサ搭載
- SDRAM (HY57V641620FTP-7、64Mbit)搭載
- 50MHz 水晶発振器搭載
- 5V 電源で給電、電源スイッチと電源指示 LED 付き
- R_C キー、押下して EPCS4 からソースを読んで実行開始する
- 1085-3.3V/1117-1.2V レギューレータ搭載
- JTAG I/F、SOF ファイルをダウンロードする。直接 FPGA に書き込んで、速度は速いですが、電源切れたらなくなる。デバッグする時に利用するのをお勧め。
- ASP I/F、POF ファイルをダウンロードする。コンフィギュレーションデバイス EPCS4 に書き込む。速度は JTAG より遅いですが、電源切れても保持する。最後のプログラム或いは電源を再起動が必要な場合利用する。※書き込み終了したら、電源を切って、ケーブルを抜けてから、正常に次の操作が出来る。
- ４本制 RS232



1	VCC3.3v	2	VCC3.3v

3	VCC5v	4	B4_7
5	B4_23	6	A9
7	SRAM ジャンパ	8	GND

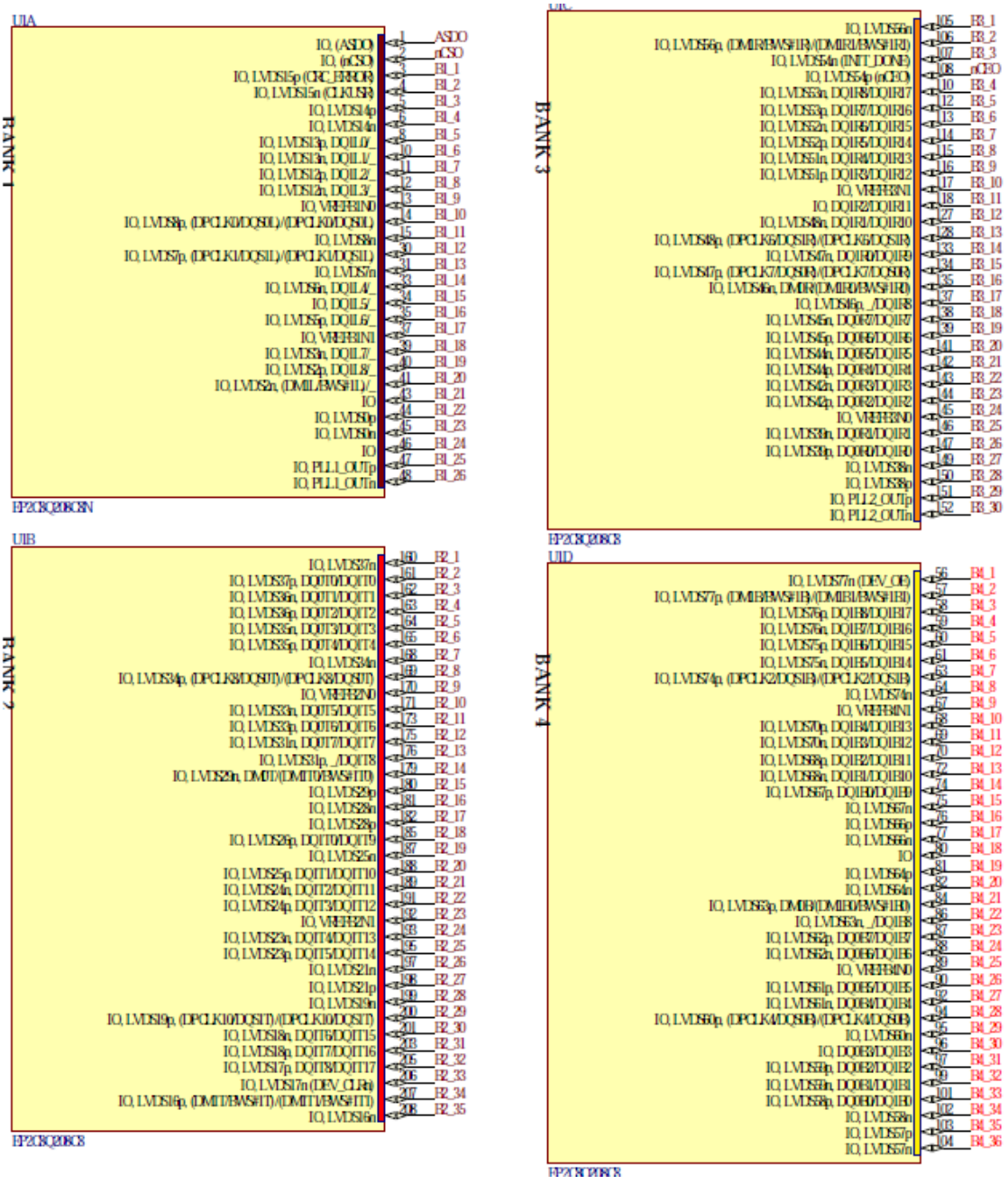
中の２、４、６、８番のピンが４本制の RS232 の端子として利用する。

また 6 番の A9 は SRAM にも使う。なので、RS232 と SRAM 同時には利用できない。

- SRAM ジャンパ、上記５と７番をショートした場合 SRAM 機能を利用できる。
- ユーザ LEDx1
- ユーザボタン x1、Reset キーと複用
- すべての IO を 2.54mm 拡張ピンヘッダで引き出されている
- コンフィギュレーションデバイス EPCS4(4Mbit)搭載
- SRAM（IS61LV25616AL 256kx16b）搭載
- I2C（AT24C02）搭載
- 外形寸法：100×62(mm) ※突起物は除く
- 回路図を提供しております
- サンプルのソースコードを提供しております

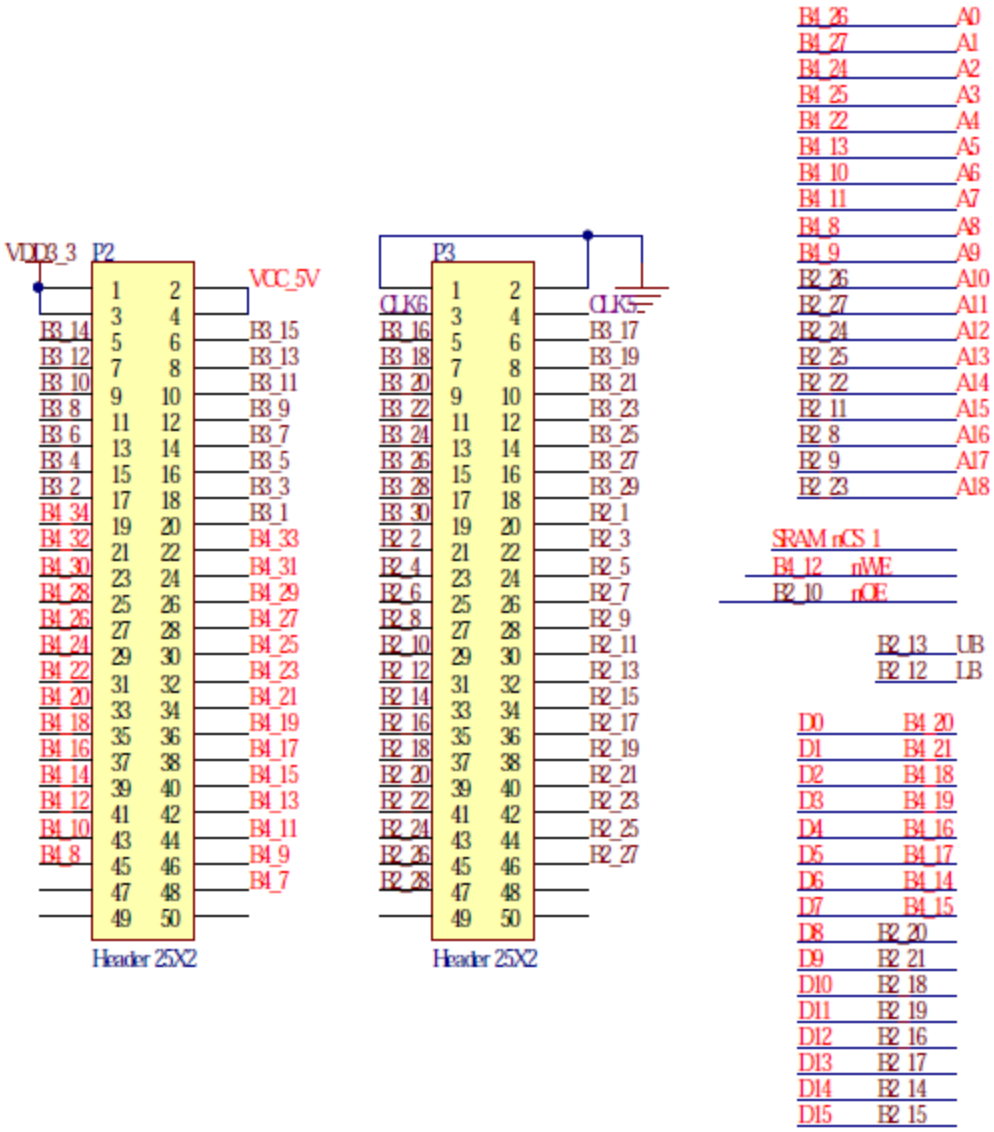
1.2 ハードウェアの構造

1.2.1 四つの BANK



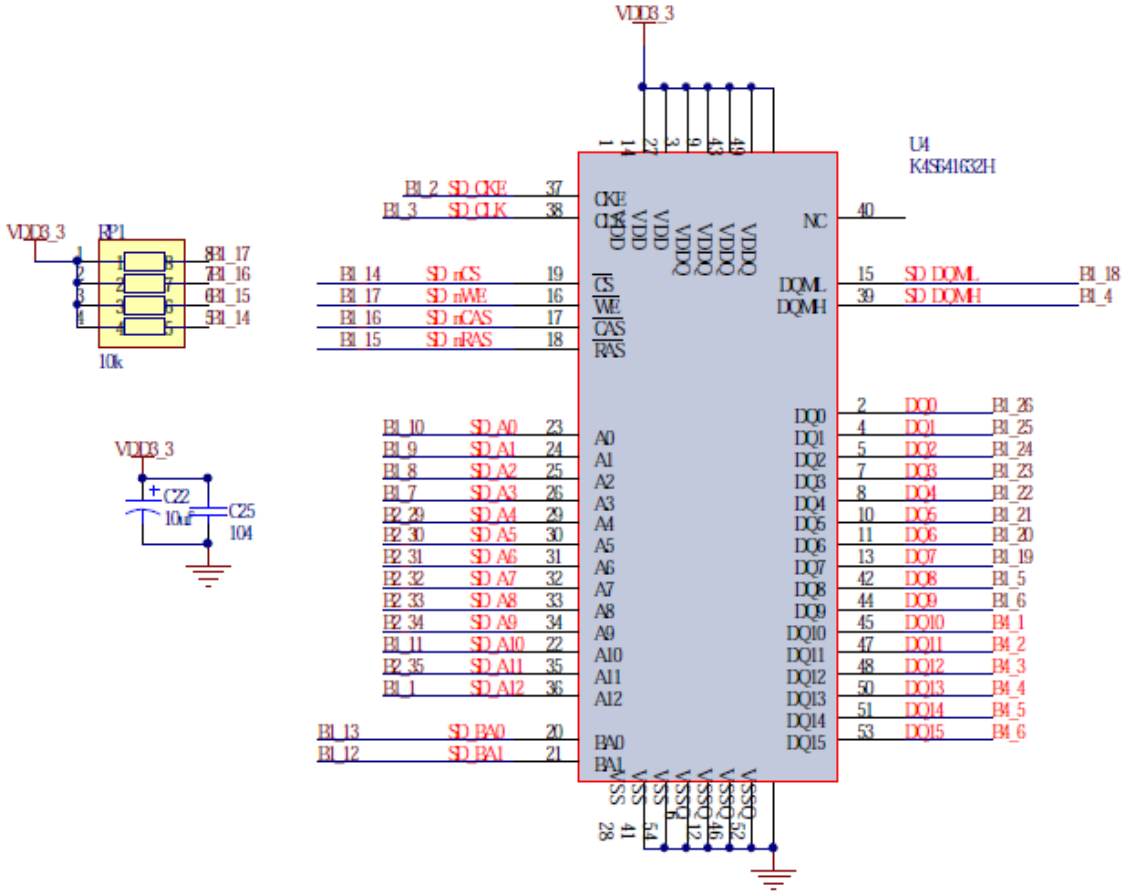
1.2.2 拡張ピンヘッダ及び SRAM インタフェース

引き出されているピンは下記の図の通り：



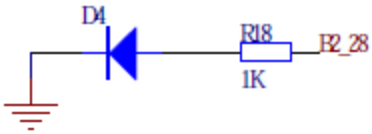
利用していないピンと電源関連ピンを除いて、：100 - 6（NC）- 6 （power） = 88個のIOピンが利用できる。その内CLK5 とCLK6はInputピンで、その他は全てInput/Output両方使えるピンになっている。図の右側はIS61LV25616AL，512kのSRAMのピン配置。

1.2.3 メモリ SDRAM の回路



HY57V641620FTP、64Mbit 容量、アドレス線は A0~A11、A12 は今後拡張用。

1.2.4 ユーザ LED（D4）の回路

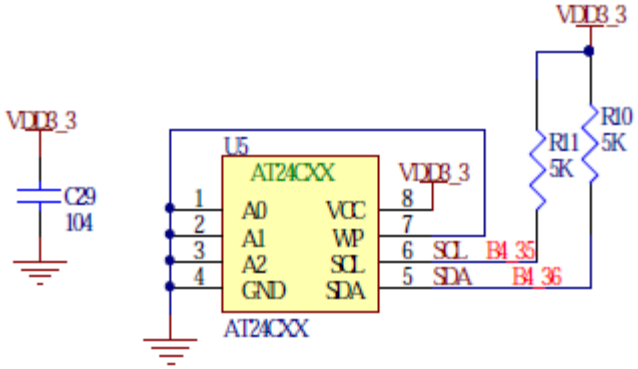


ピン配置：

```
# led on core board
set_location_assignment PIN_199-to led
```

信号	ピン	機能
LED	199	LED

1.2.5 I2C インタフェース

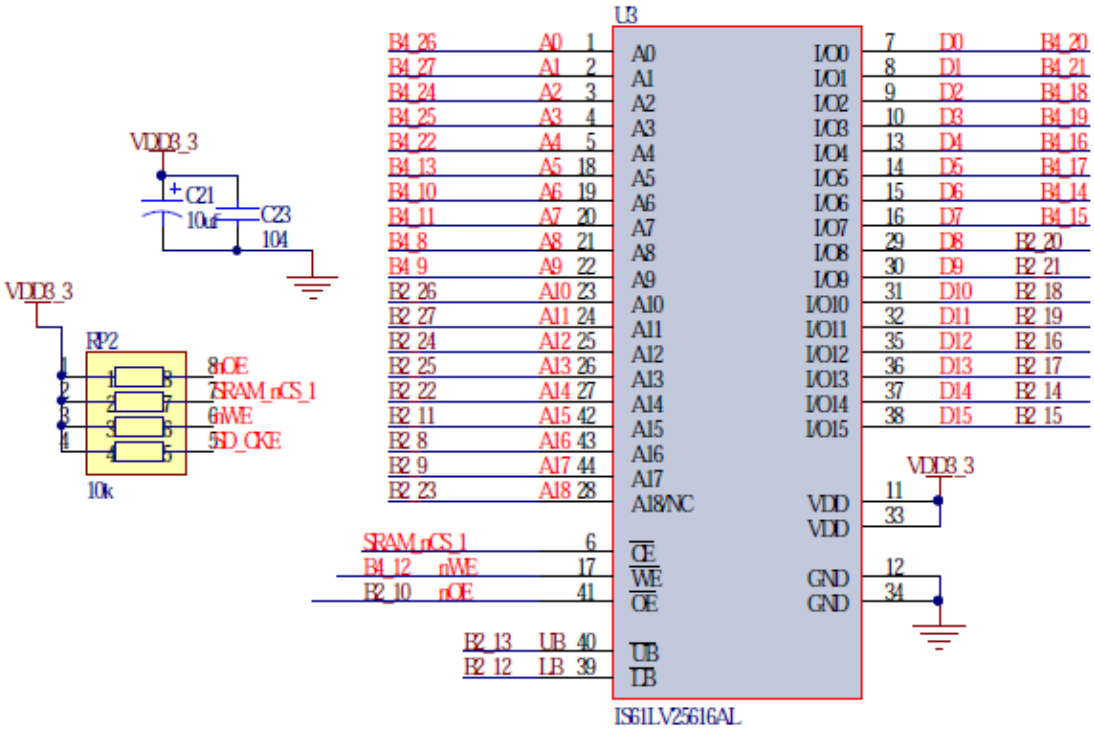


ピン配置：

```
# AT24C02 on core board
set_location_assignment PIN_103 -to SCL_I2C
set_location_assignment PIN_104 -to SDA_I2C
```

信号	ピン	機能
SCL_I2C	103	AT24C02 シリアル通信クロック信号
SDA_I2C	104	AT24C02 シリアル通信データ信号

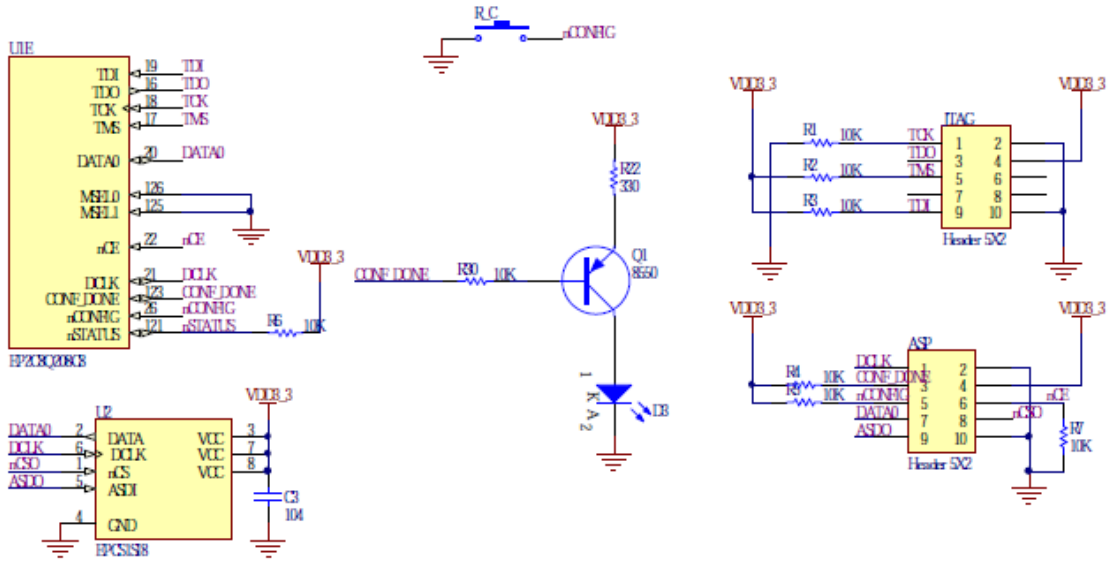
1.2.6 SRAM インタフェース



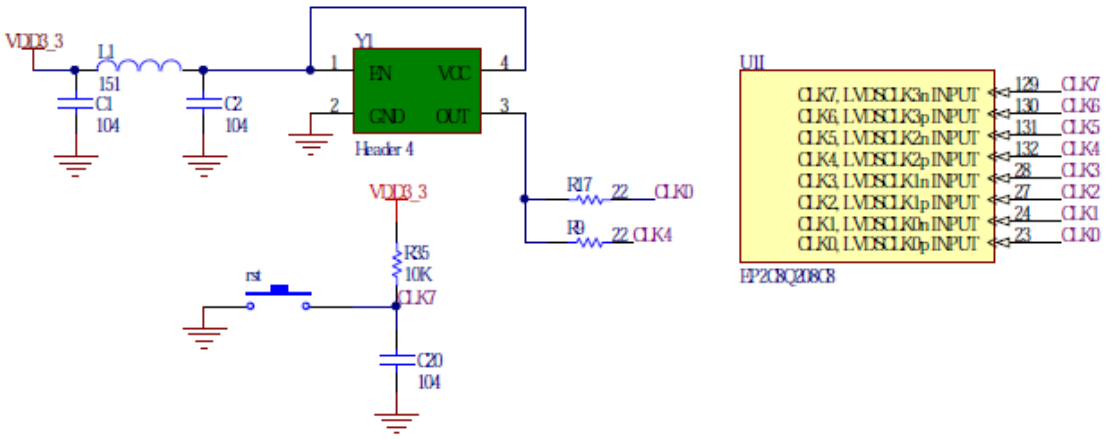
RAMを拡張すると共にNios IIをもっと良くサポートする為、SRAMインタフェースを提供し

ている。IS61LV25616ALで実現している。アドレスラインが18本で、A18は１MのSRAMと交換性を持つためである。その他、データラインが16本、制御ラインが5本で、合わせてIOの39個を使っている。256x16bKbitで512kBの容量である。

1.2.7 コンフィギュレーションの回路



1.2.8 クロック及び RESET 回路



本ボードは 50MHz 水晶発振でシステムにクロックを提供している。ユーザーに 8 個のクロックを用意している。全部普通のインプットピンとして利用できる。設計上では下記の様に設定している：

１、CLK0 と CLK4 はシステムにクロックを提供し、直接 50MHz 水晶発振と接続している。

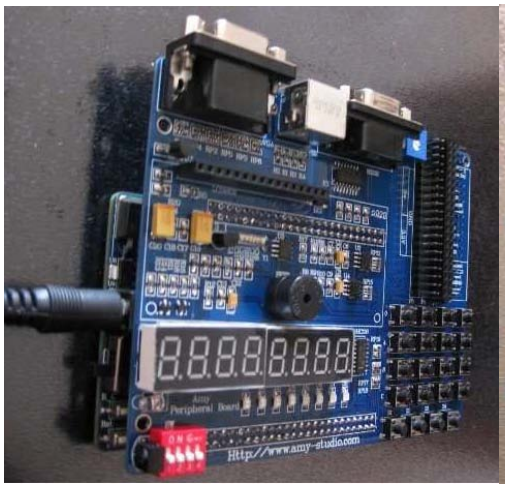
２、CLK7 はユーザのプログラミングによって RESET として利用できる。

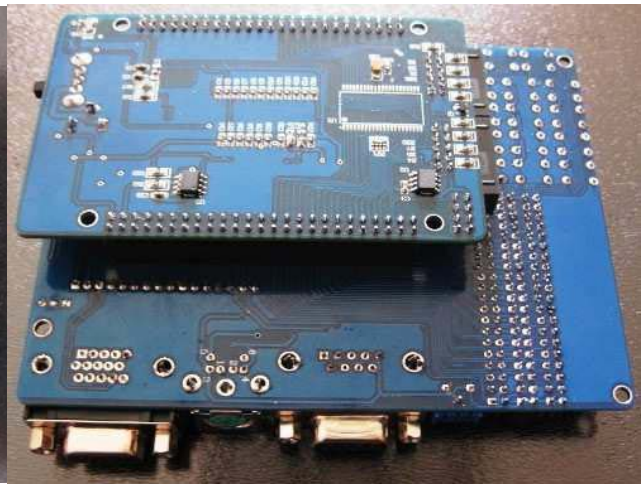
３、CLK5 と CLK6 は引き出されていて、インプットピンとして利用できる。

４、CLK1、CLK2 と CLK3 は引き出されてない。

1.3 CPLD/FPGA の実験用 I/F ボードとの接続

イメージ：





引き出しているピンリスト：

P2			P3	
3.3v	5v		GND	GND
3.3v	5v		130	131
133	134		135	137
127	128		138	139
117	118		141	142
115	116		143	144
113	114		145	146
110	112		147	149
106	107		150	151
102	105		152	160
99	101		161	162
96	97		163	164
94	95		165	168
90	92		169	170
88	89		171	173
86	87		175	176
82	84		179	180
80	81		181	182

76	77		185	187
74	75		188	189
70	72		191	192
68	68		193	195
64	67		197	198
NC	63		199	NC
NC	NC		NC	NC

1.4 サンプルソースについて

Example_ EP2C8.zip に下記サンプルソースは含まれている。

1.4.1 sdram_basic

NIOS II の基本的なテストプログラム

1.4.2 Nios_example

seg7x8_test

7SEGMENT 動的スキャン表示テスト

uart_test

UART 発送、受信テスト

ボーレート：9600

n5110_lcd_test

nokia 5110 lcd 表示テスト

max7219_test

Max7219 で SEG7 ドライブするテスト

SPI 通信練習

lcd1602_test

LCD1602 液晶テスト

lcd12864_test

lcd12864（st7920）液晶テスト

key_led_test

led_test

led 点滅テスト

key_irq_led_test

キー割り込みで LED を制御するテスト

その他：

Amy_S_ip

簡易 ip テスト

1.4.3 Logic_verilog

turn_on_led

　　LED 点灯

sw_led

　　DIP で LED 制御

rider_led

　　跑马灯

water_led

　　ウォーターLED

key_led_without_debounce

　　タッチ SW で LED 制御（手ぶれ処理なし）

key_led_with_debounce

　　タッチ SW で LED 制御（手ぶれ処理あり）

seg7x8_dynamic_disp

　　7SEGMENT 動的表示

matrixKeyboard_seg7

　　キーマトリクスと 7SEG 表示

beep_test

　　ブーザテスト

beep_matrixKeyboard

　　周波数よりの簡易ブーザテスト

lcd1602_test

　　LCD1602 表示

lcd1602_clock

　　簡易クロック、LCD1602 表示

vga_color_slip

　　VGA カラースクリーン表示

vga_char

　　VGA キャラクター表示

uart_tx_test

　　シリアル通信送信テスト

uart_rx_test

　　シリアル通信受信テスト

ps2_keyboard_test

　　PS2 キーボードテスト

ds18b20_seg7

DS18B20 温度検出、7SEG 表示テスト

1.4.4 sram_25616

SRAM テスト

1.4.5 Logic_vhdl

VHDL プログラム例（LED 点灯、ウォーターLED）

1.4.6 EP2C8 ボードの LED テスト

ボードの簡易テスト

第二章 開発ツールのインストール

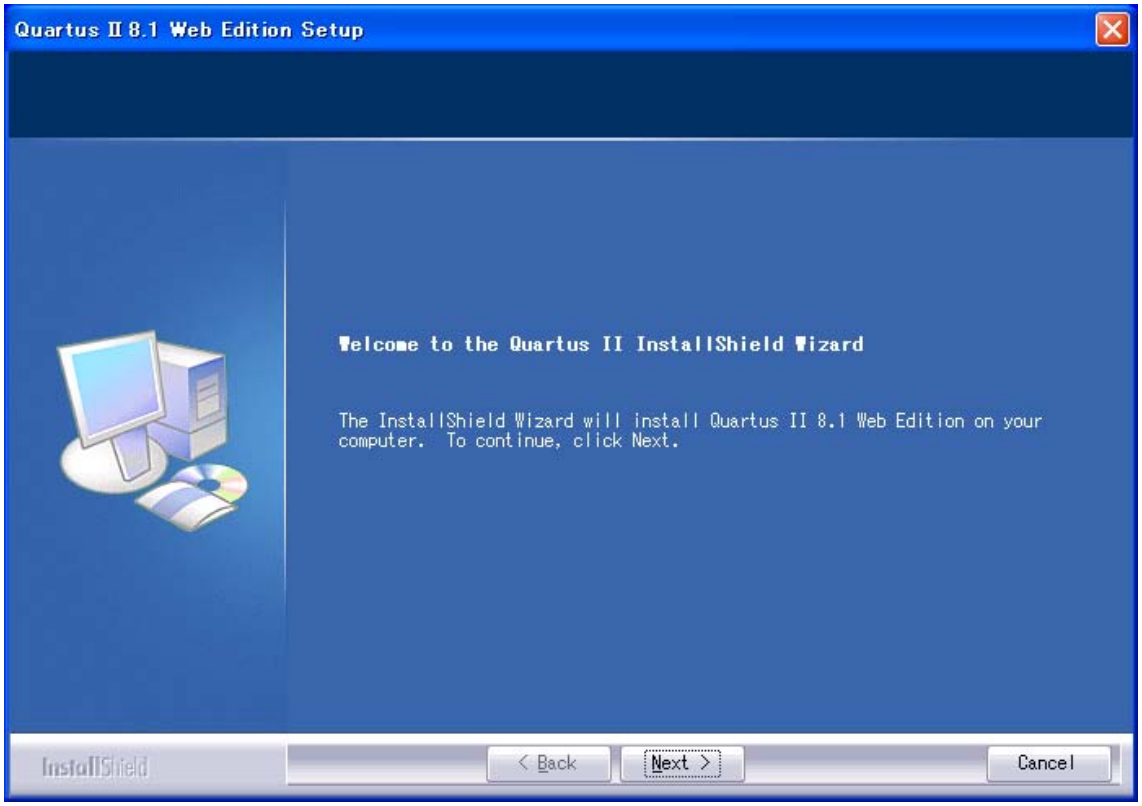
CPLD/FPGA の開発には、ALTERA から Quartus II Web Edition という無償版のツールが公開されているのでこちらを利用します。Quartus II には別に製品版があり、Web Edition は使用できるデバイスなどに制限がありますが、MAX II と Cyclone II に関しては、どのデバイスも使用できるのでまったく問題ありません。Quartus II Web Edition は、総合開発環境になっており、このソフトウェアだけで、ソース・エディタや I/O ピンのアサインメント、論理合成、デバイスの書き込み用のプログラムなど、CPLD/FPGA の開発に必要な機能がすべて含まれています。また、Nios II エンベデッド・デザイン・スイートは Nios プロセッサー用の開発ツールです。

Quartus II Web Edition と Nios II エンベデッド・デザイン・スイートのダウンロードは、次の URL から行うことができます。

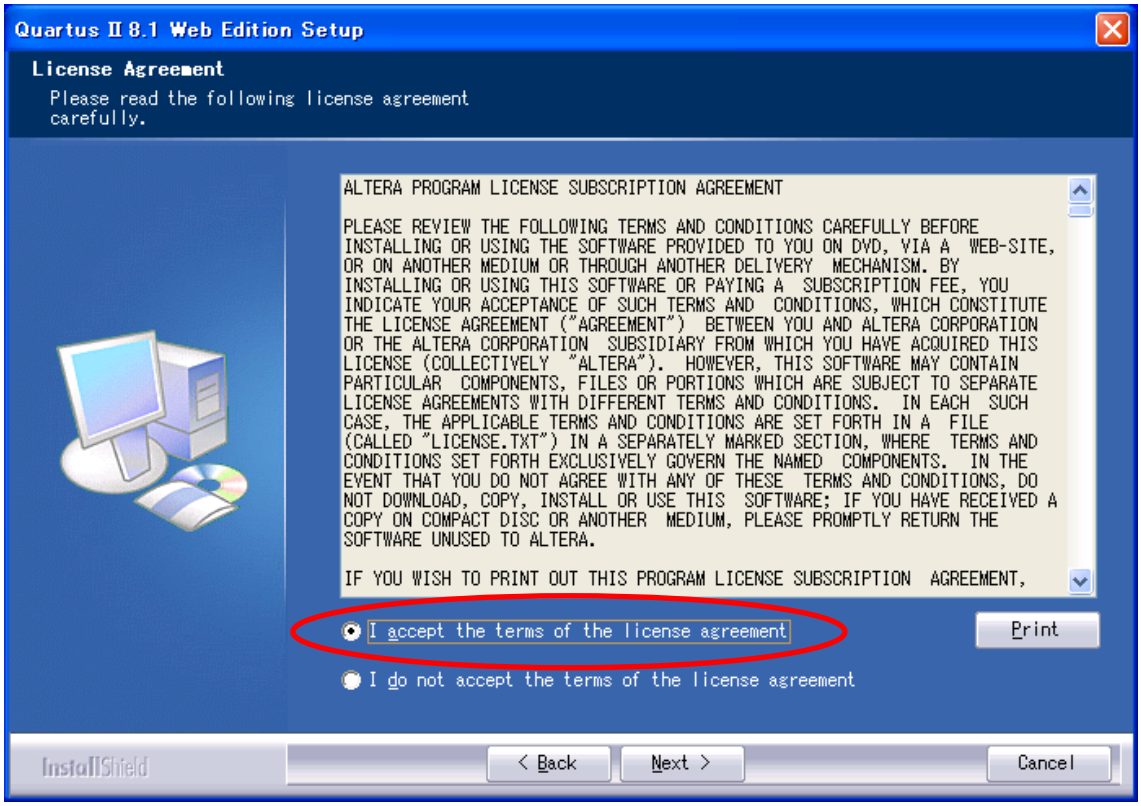
http://www.altera.co.jp/support/software/download/nios2/dnl-nios2.jsp

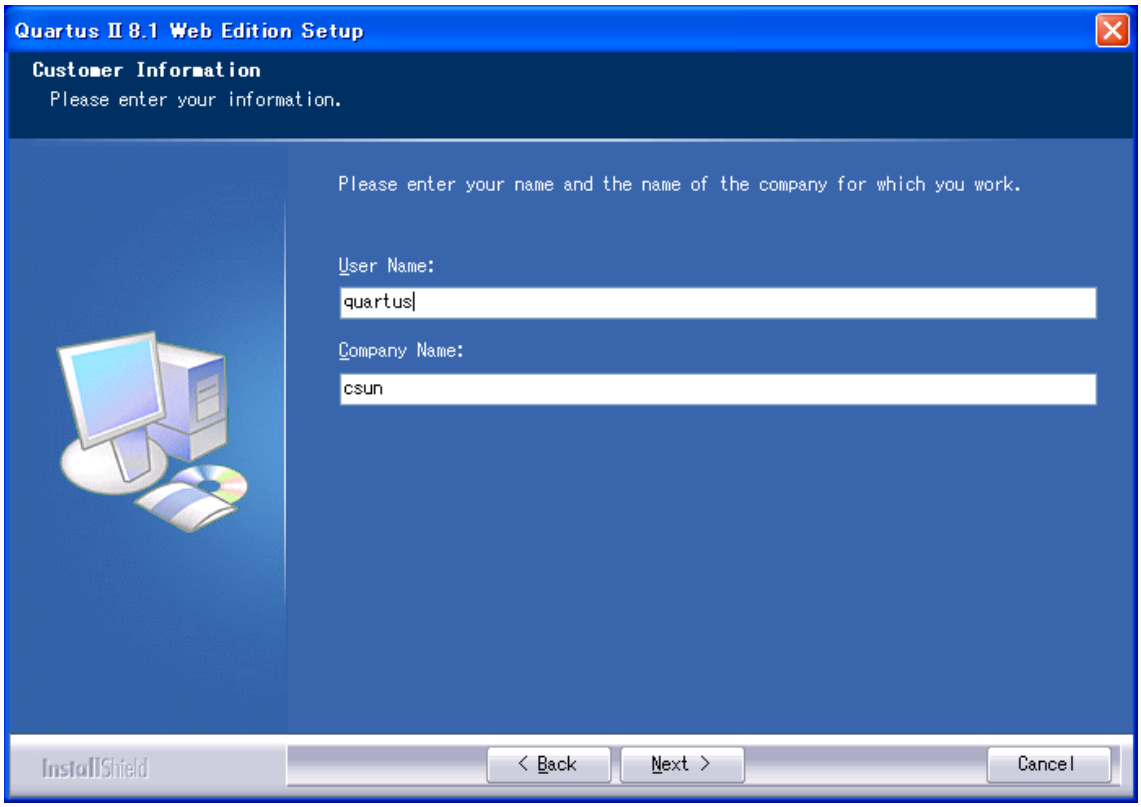
なお、ダウンロードする際は、最初に ALTERA のページにサイン・インを行い、ユーザ情報を登録する必要があります。本章には v8.1 でインストールの手順を説明します。インストールした後、ライセンス・ファイルが不要です。

2.1 Quartus II Web Edition をインストールする

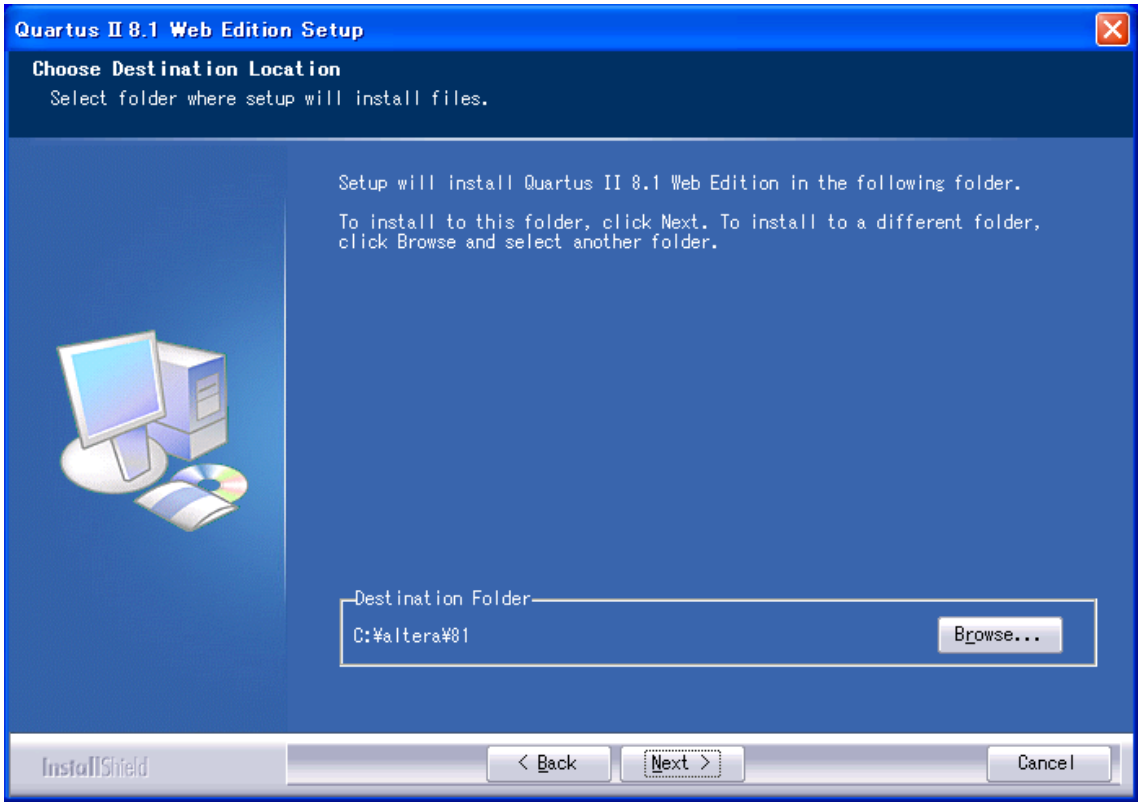


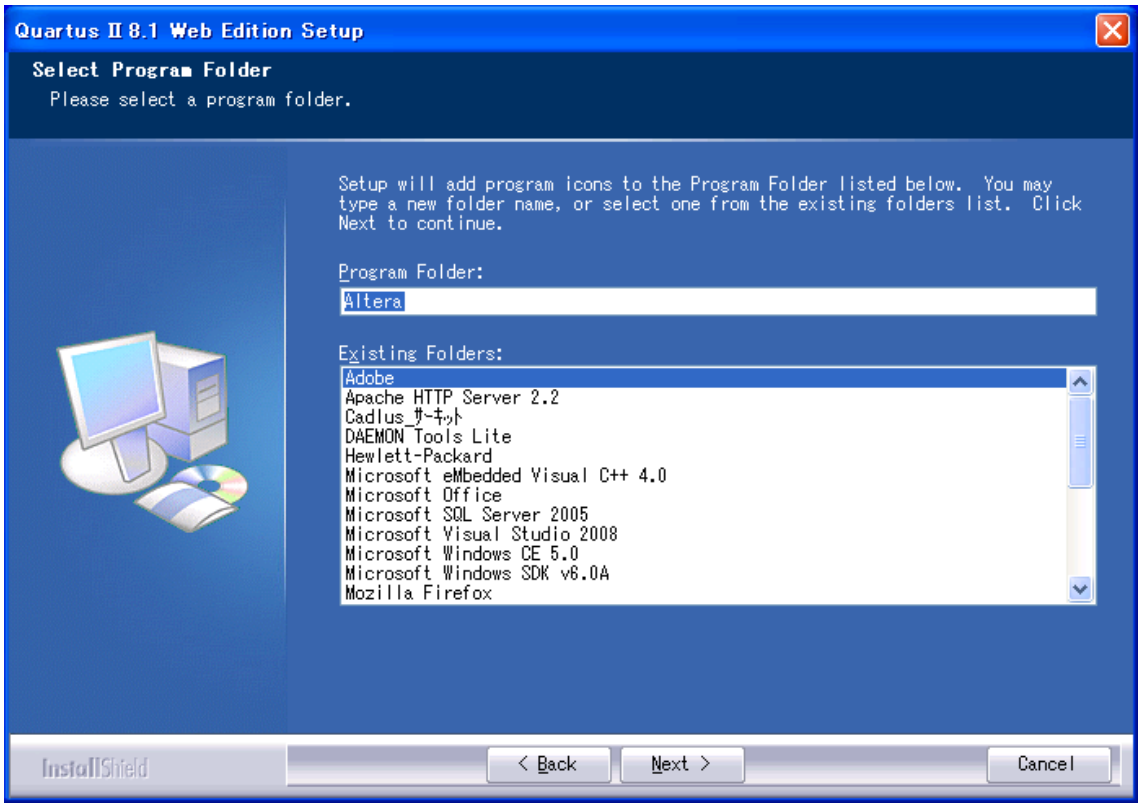
「Next」ボタンを押すと、英文のライセンスが出てきます。同意できる場合は、「**I accept the terms of the license agreement**」を選択して、「Next」ボタンを押します。



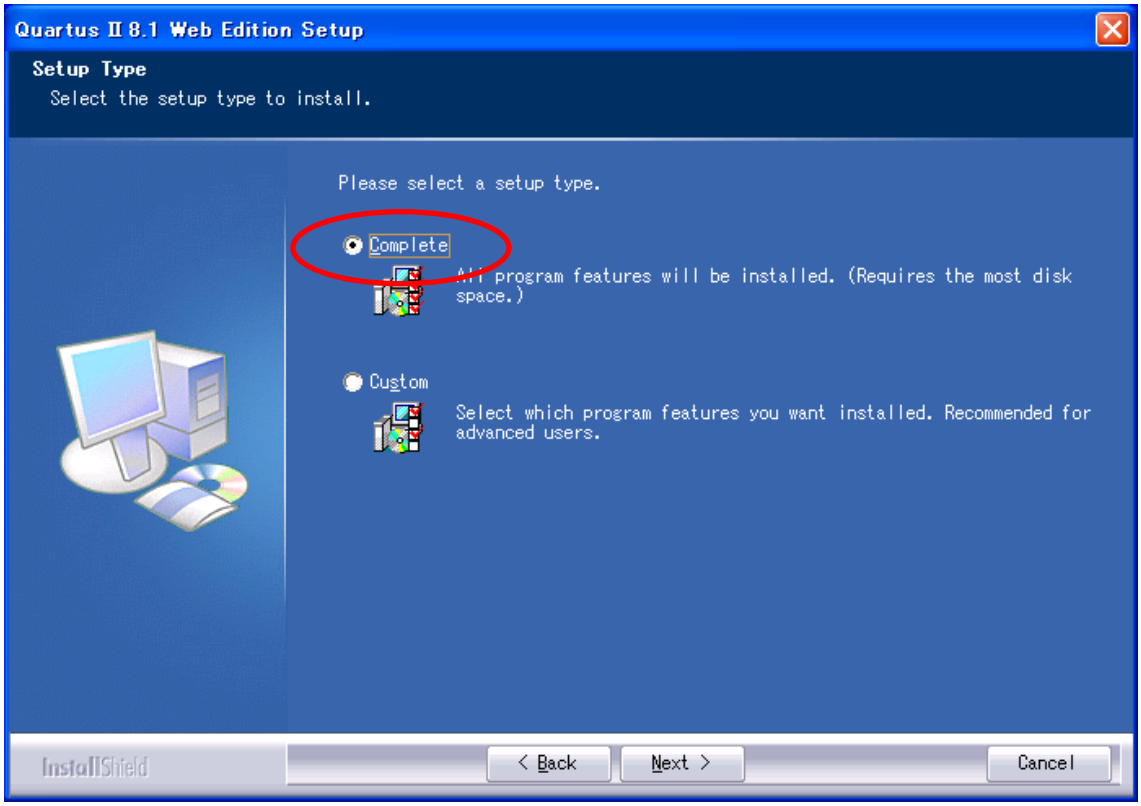


使用者の名前と所属会社名を入力するダイアログが表示されます。名前は半角のアルファベットで入力しましょう。

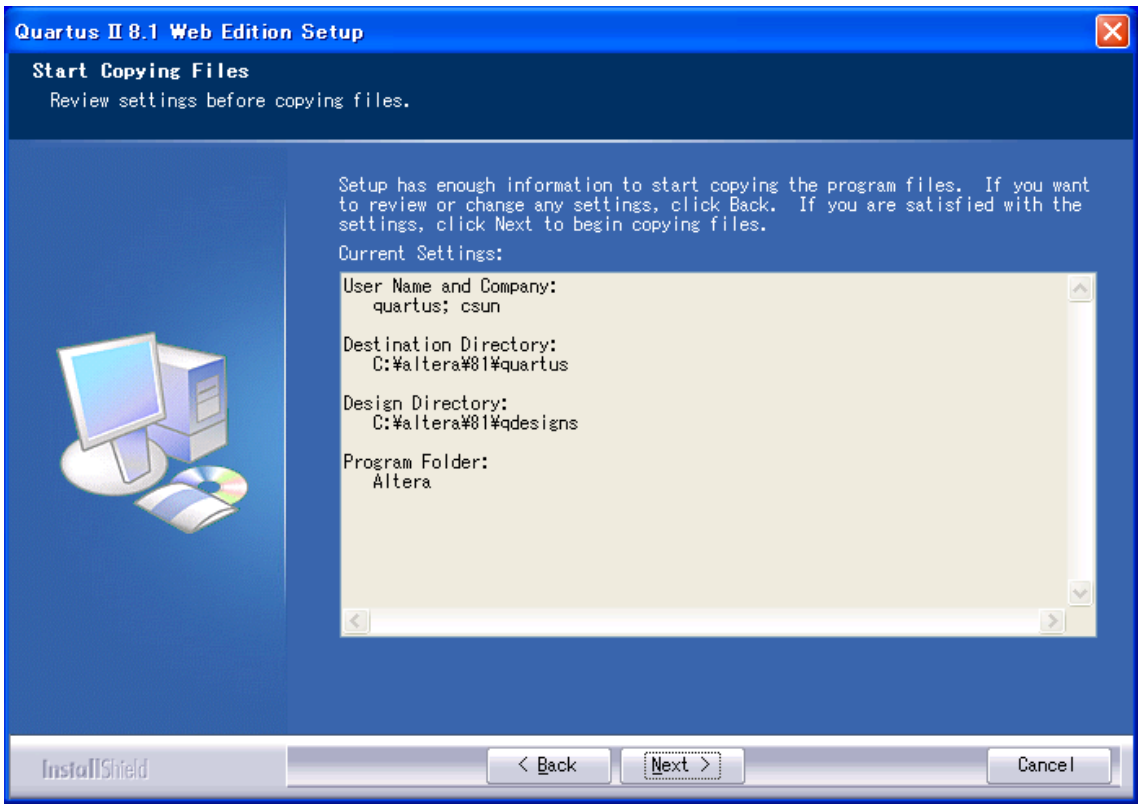




インストール先フォルダを変更せず、そのまま進んでください。

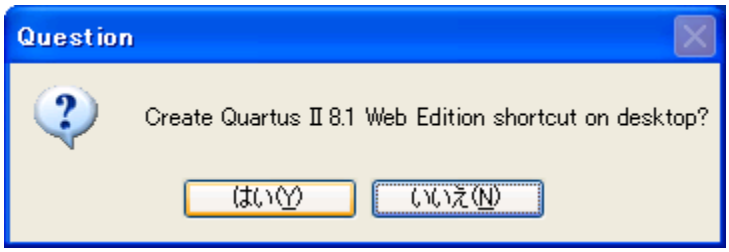


「Complete」を選択してください。

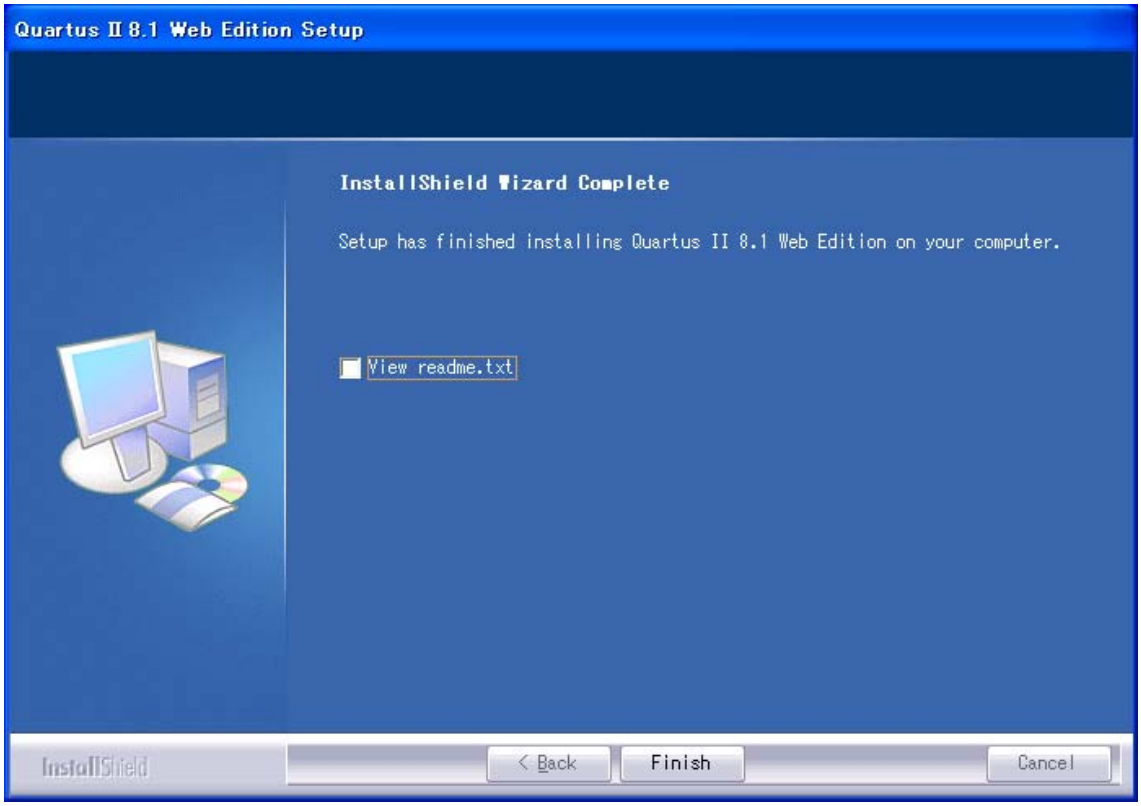


間違いがないかどうか確認し、問題がなければ「Next」を押します。

インストール中の画面です。

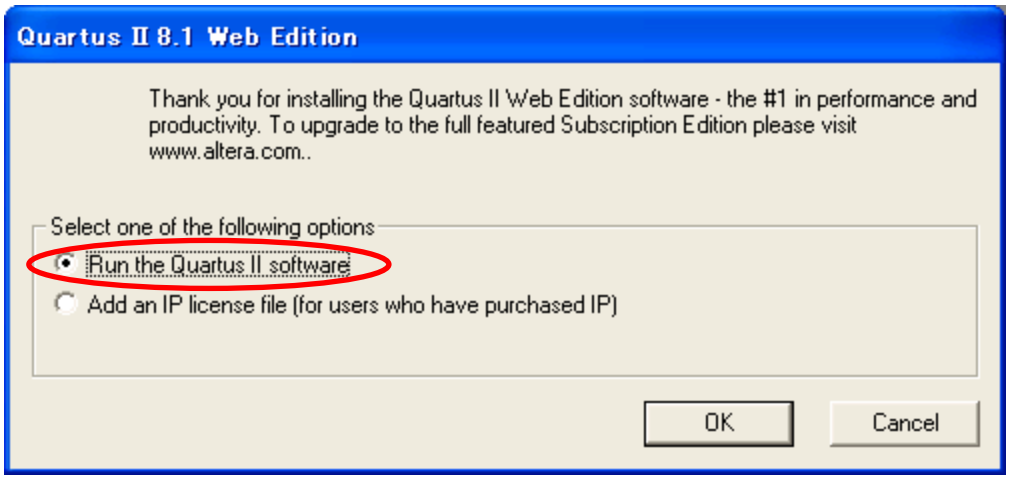


インストール完了すると、ショートカットをデスクトップに作るかどうか聞かれます。どちらでも選択できます。

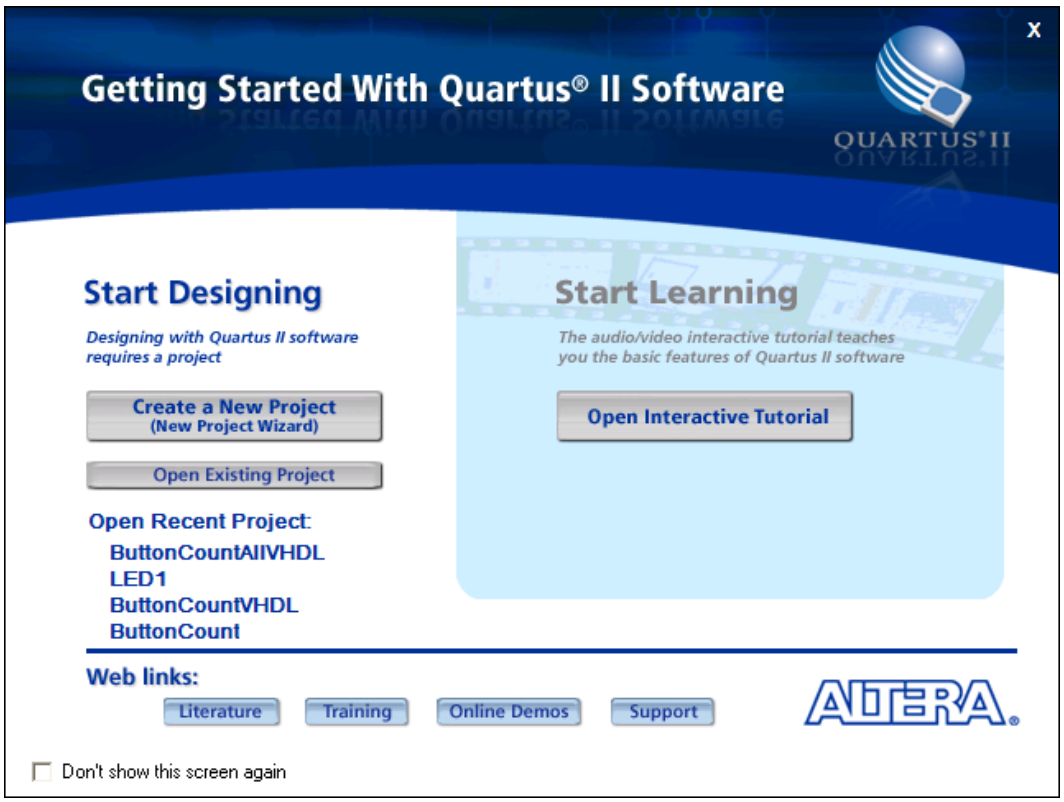


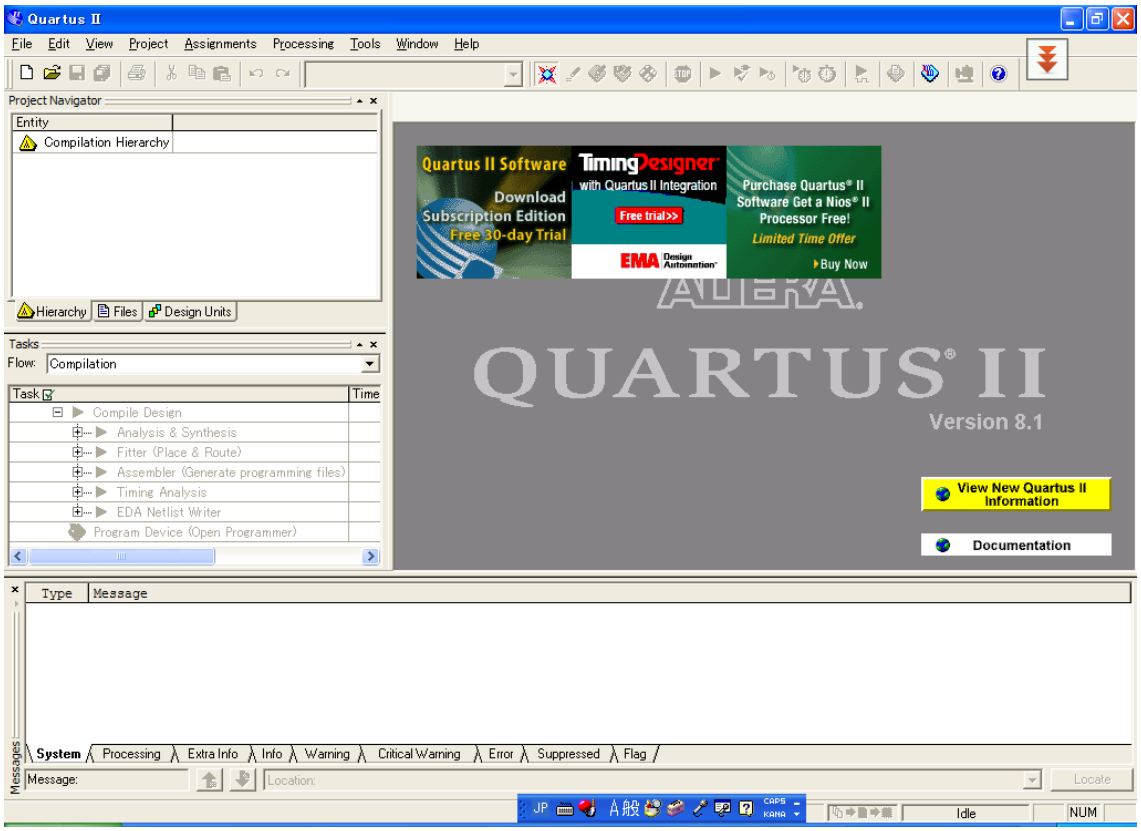
最後に「Finish」をクリックすると、ウィザードが閉じてインストールが終了します。

インストールされた Quartus II 評価版をさっそく起動してみます。一番最初に起動したときだけ、次のようなダイアログが現れ、「**Run the Quartus II software**」を選択してください。「OK」ボタンを押します。

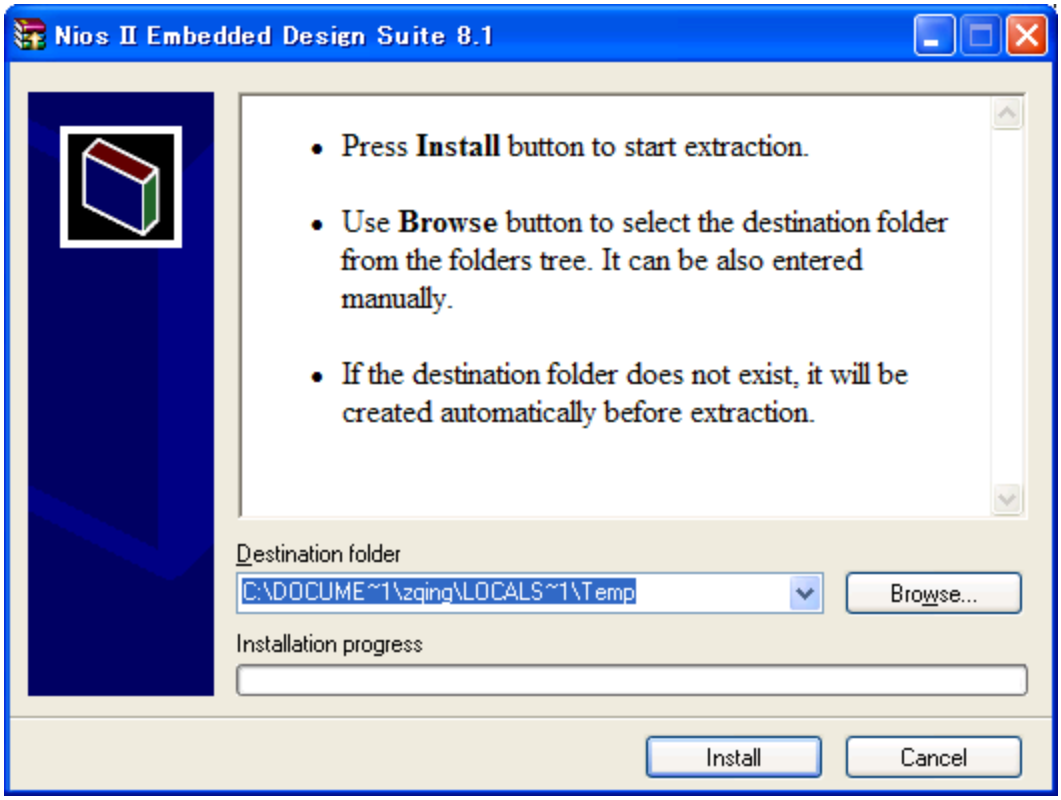


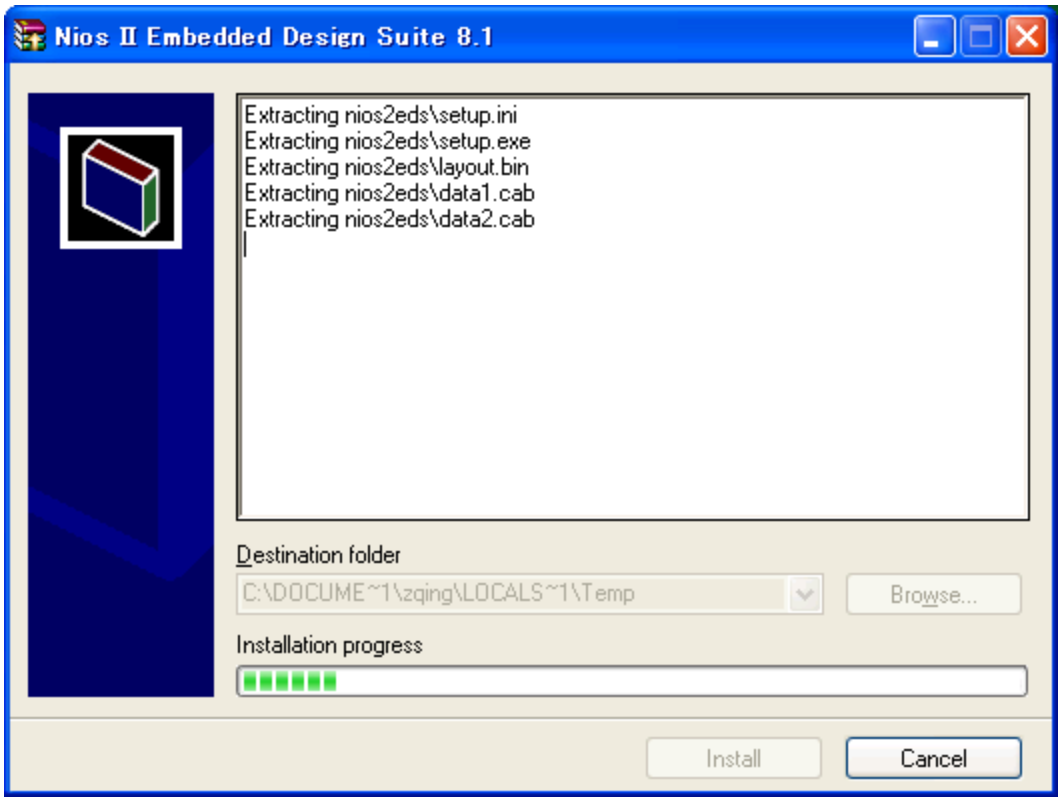
Quartus II の画面出てきます。



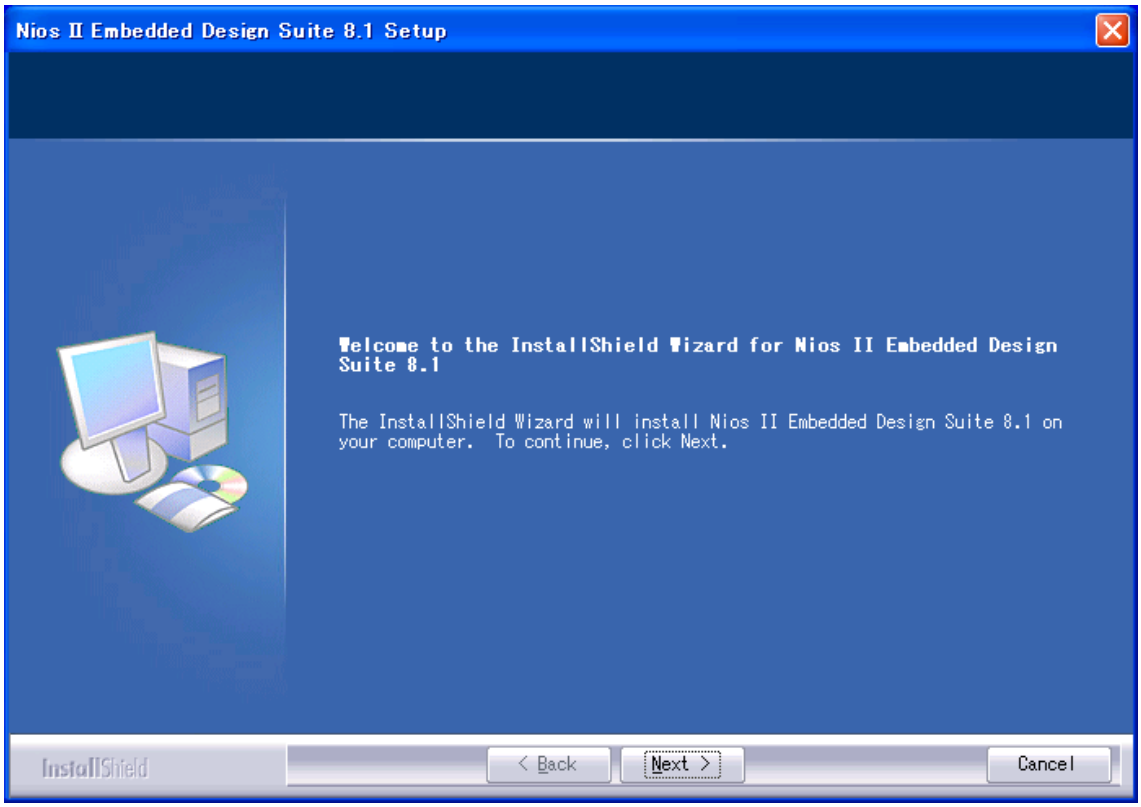


2.2 Nios II エンベデッド・デザイン・スイートをインストールする

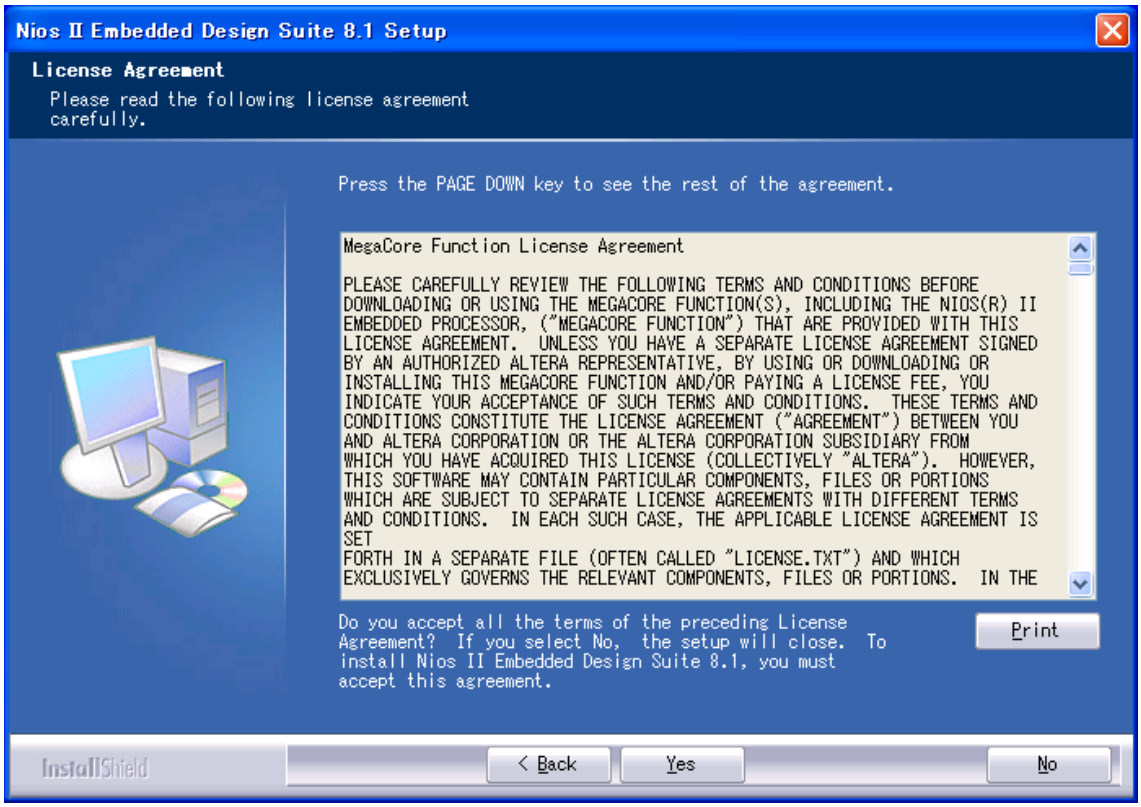


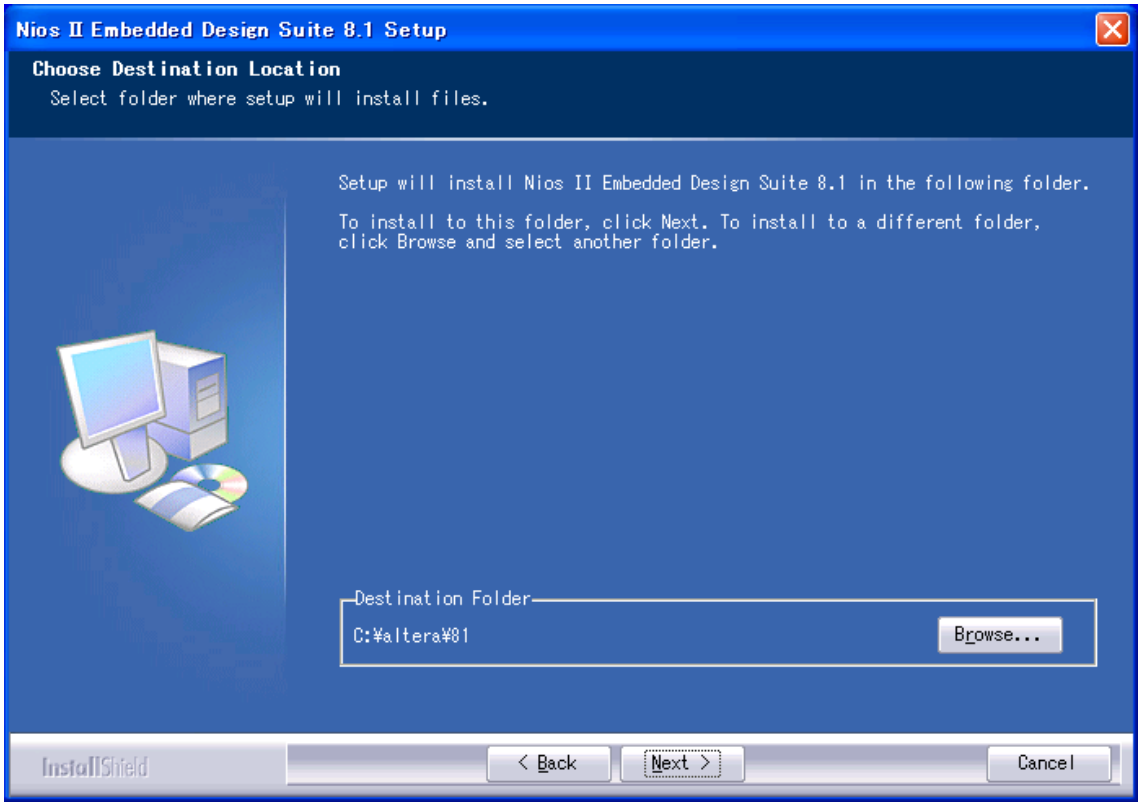


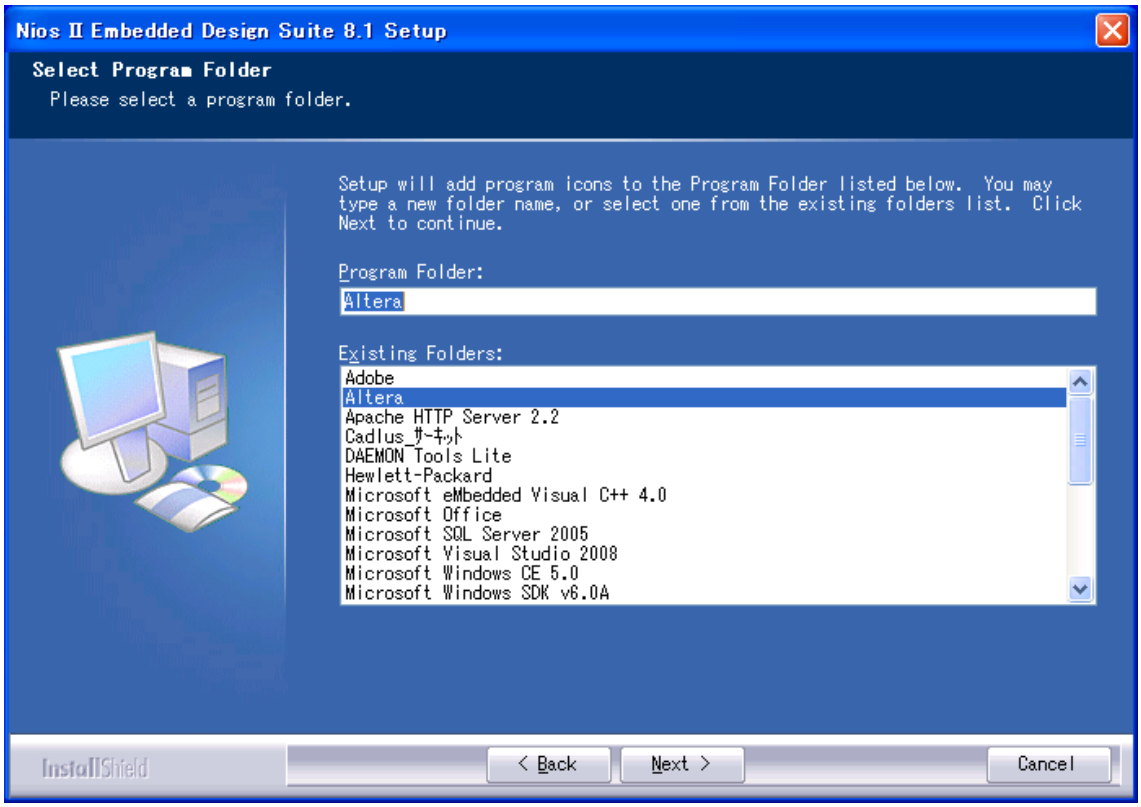
先ず「Install」ボタンを押して解凍します。「Next」ボタンを押します。



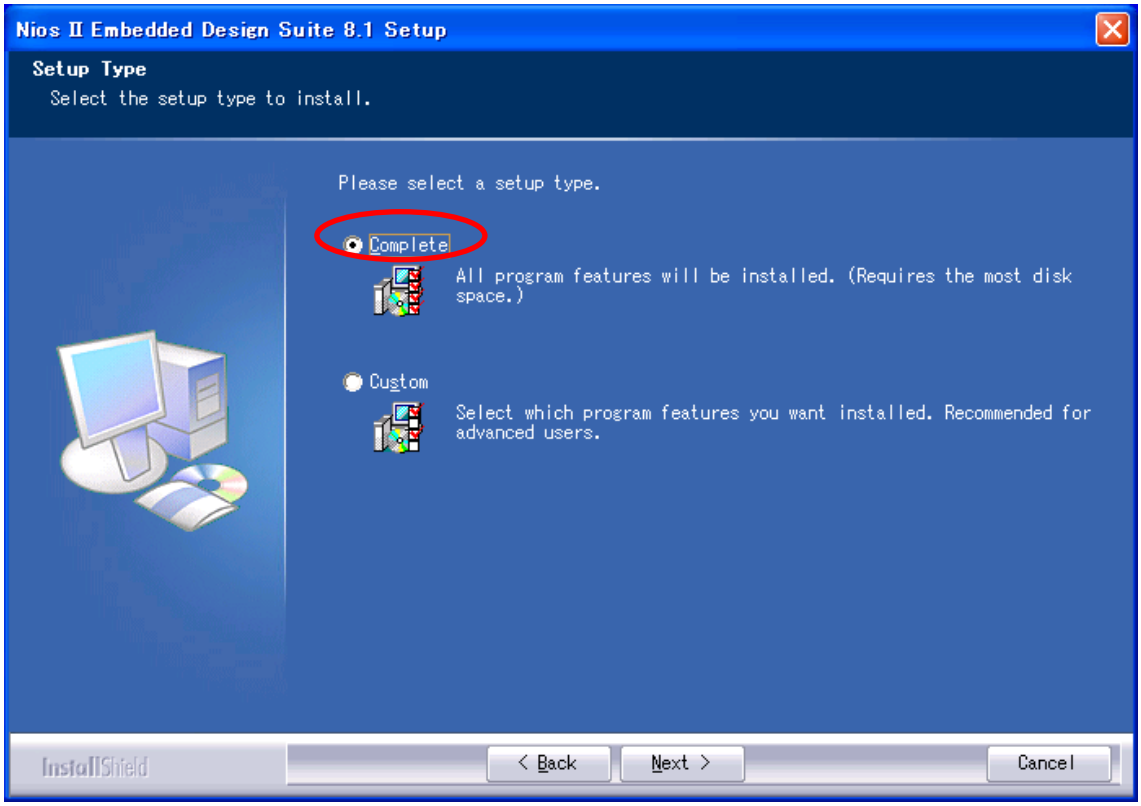
ライセンスを同意すれば、「Yes」ボタンを押します。



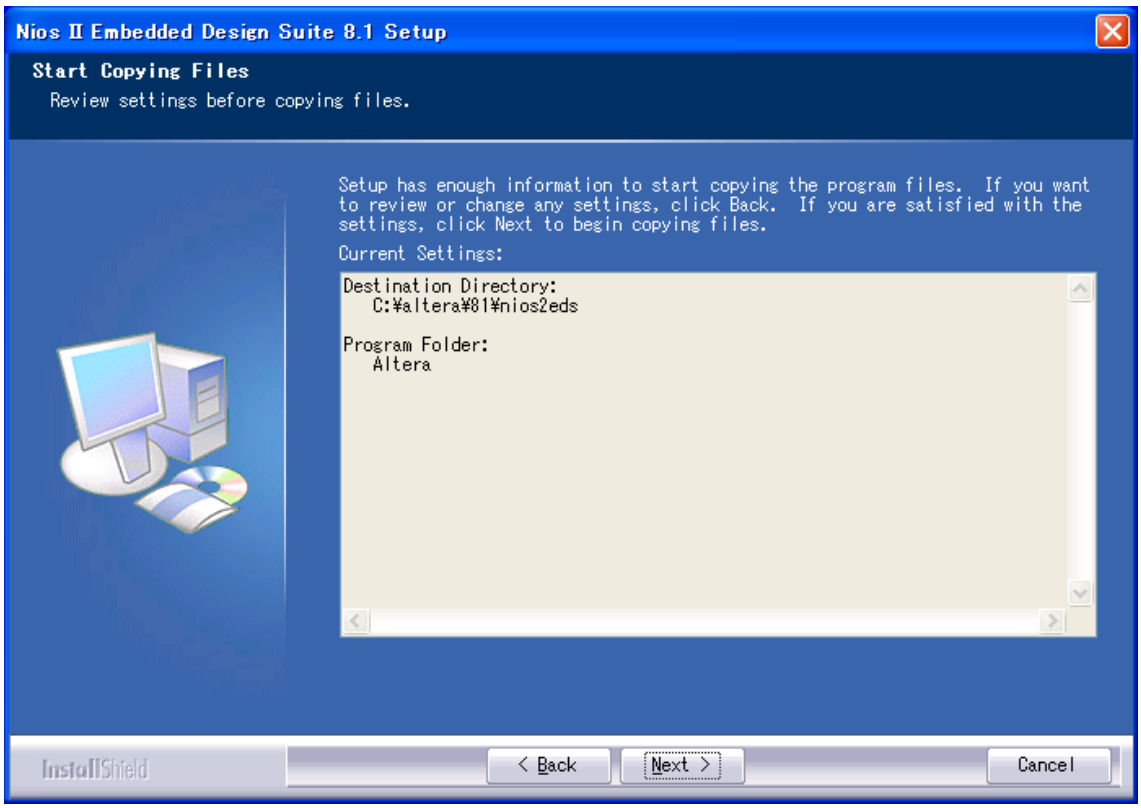




インストール先フォルダを変更せず、そのまま進んでください。

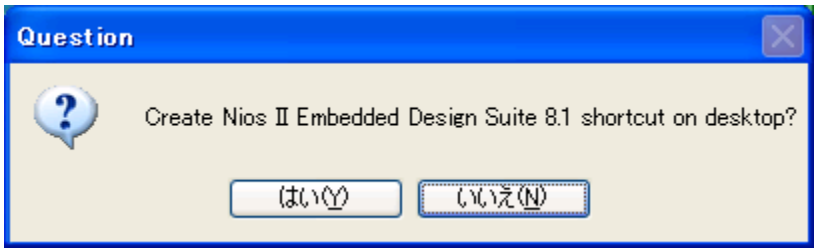


「Complete」を選択してください。

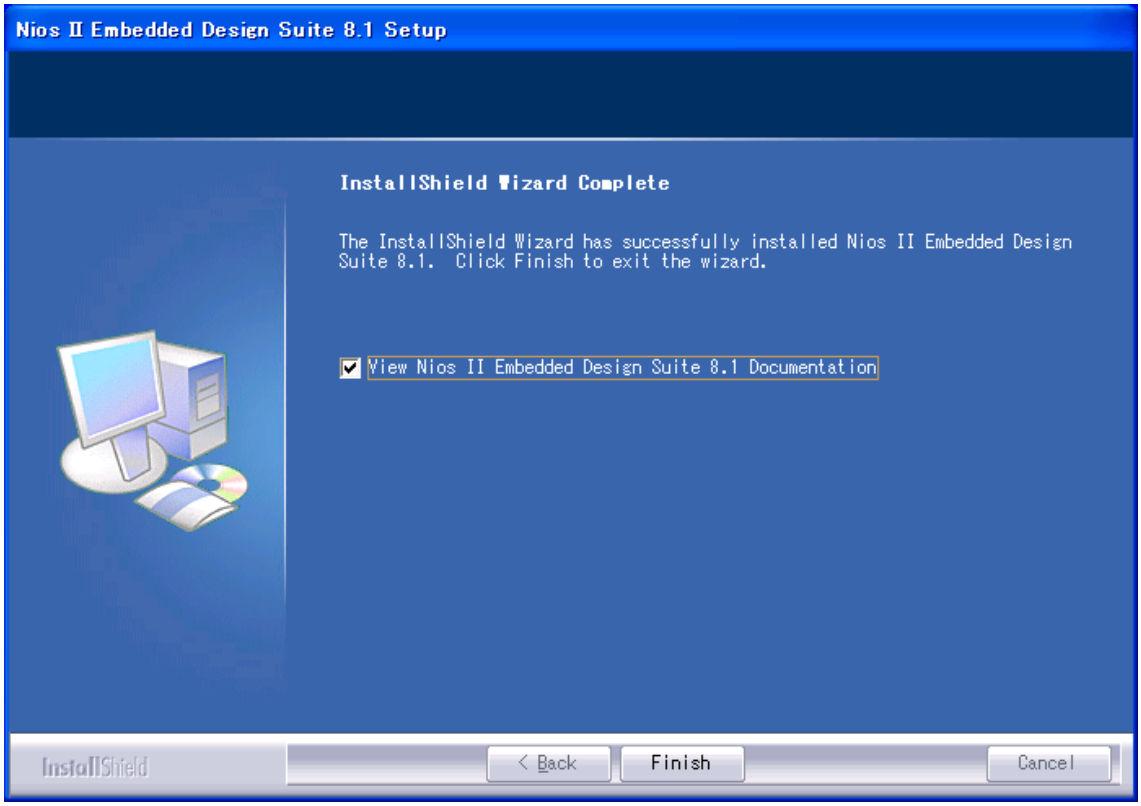


間違いがないかどうか確認し、問題がなければ「Next」を押します。

インストール中。



インストール完了すると、ショートカットをデスクトップに作るかどうか聞かれます。どちらでも選択できます。



最後に「Finish」をクリックすると、ウィザードが閉じてインストールが終了します。

第三章 Cyclone II の初体験

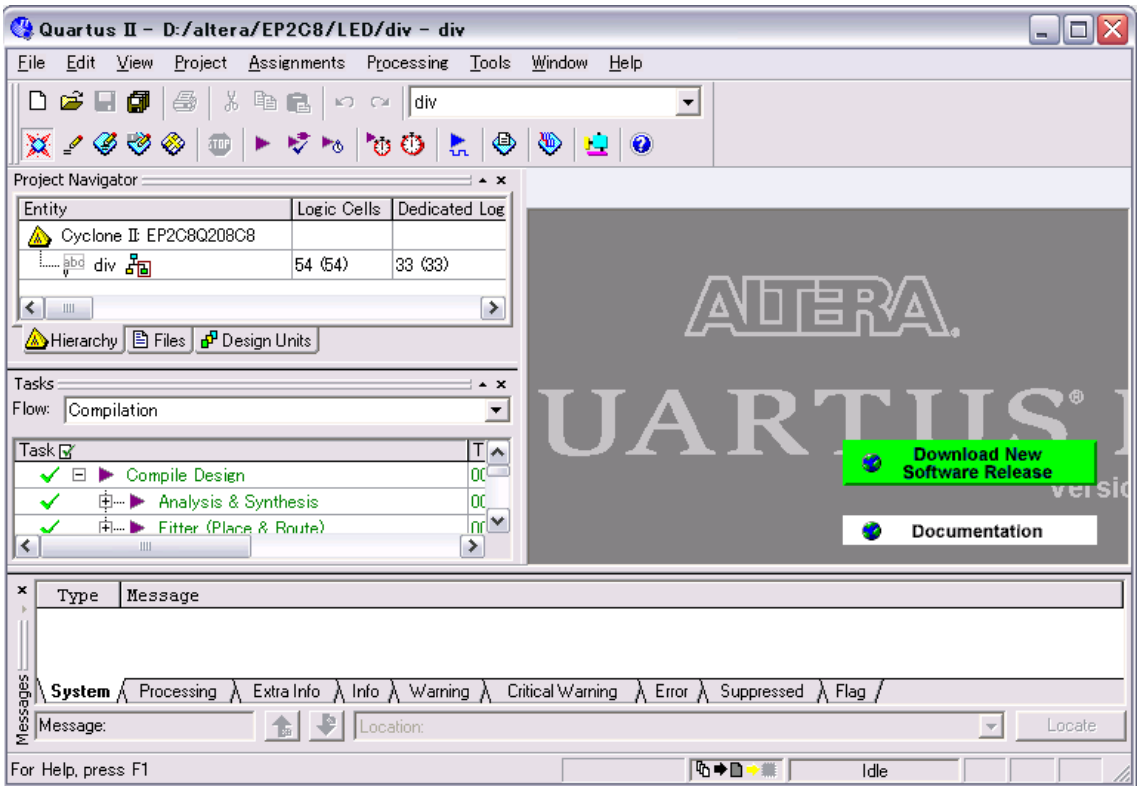
3.1 Quartus II 評価版にソースを読み込む

弊社のウェブサイトで Cyclone II 用のサンプルソース（Example_EP2C8.zip）をダウンロードできます。

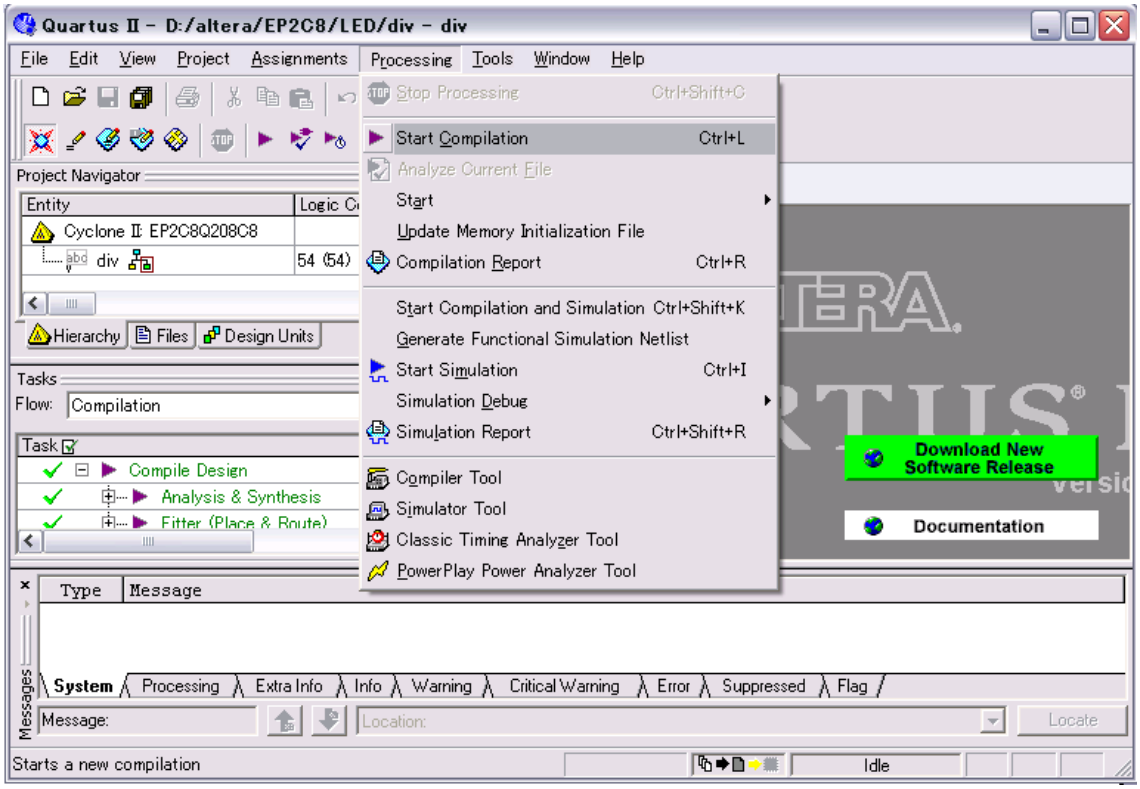
ソース・ファイルを..¥altera に展開します。その中に、幾つのサンプルがあります。具体的には 1.4 節をご参照ください。一つのサンプルを紹介します。

エクスプローラまたはマイ コンピュータを起動して、
C:¥altera¥EP2C8 ¥LED
というフォルダを開いてください。

これらの中に、名前が div.qpf、Quartus II Project File となっているファイルがあります。これをダブル・クリックすると、Quartus II が起動して、div というプロジェクトが開きます。

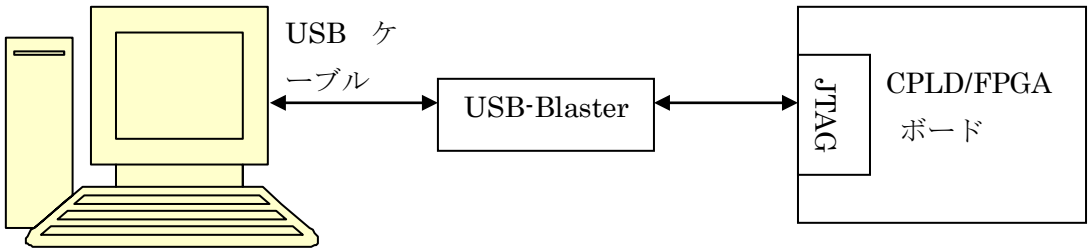


Quartus II の「Processing」メニューから「Start Compilation」を選択します。するとコンパイル処理が始まり、プロブレス・バーが働き始めます。コンパイルは数十秒で終了します。



3.2 USB-Blaster をインストールする

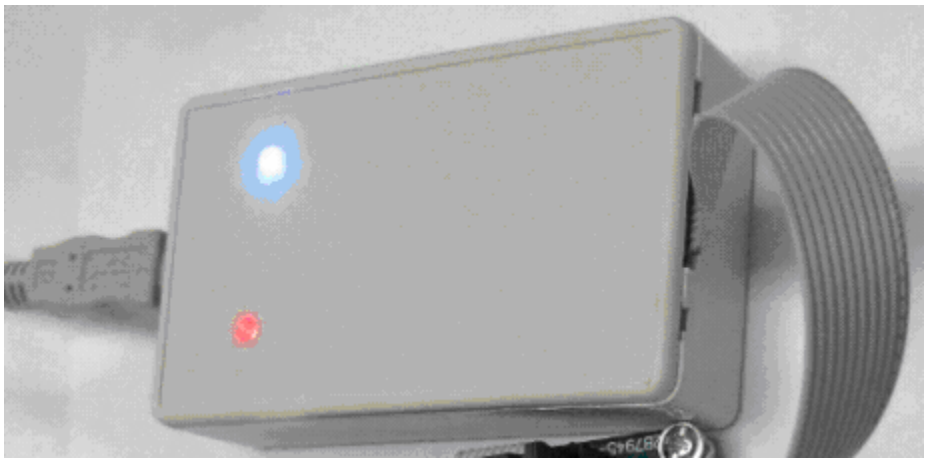
通常、MAX II/Cyclone II にコンフィグレーション・データを書き込むために、アルテラが発売している専用ダウンロード・ケーブル(ByteBlaster MV や ByteBlasterII や USB 接続タイプの USB-Blaster など)を購入しなければなりません。



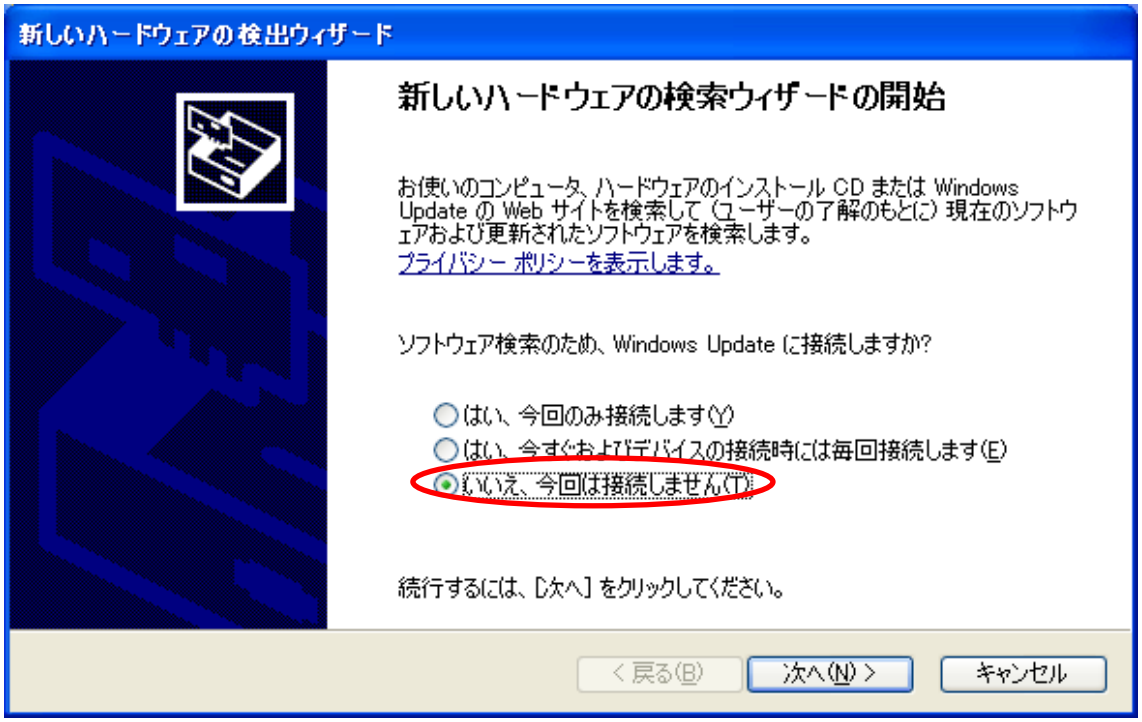
弊社は専用ダウンロード・ケーブル USB-Blaster 同等のデバイスを提供しております。

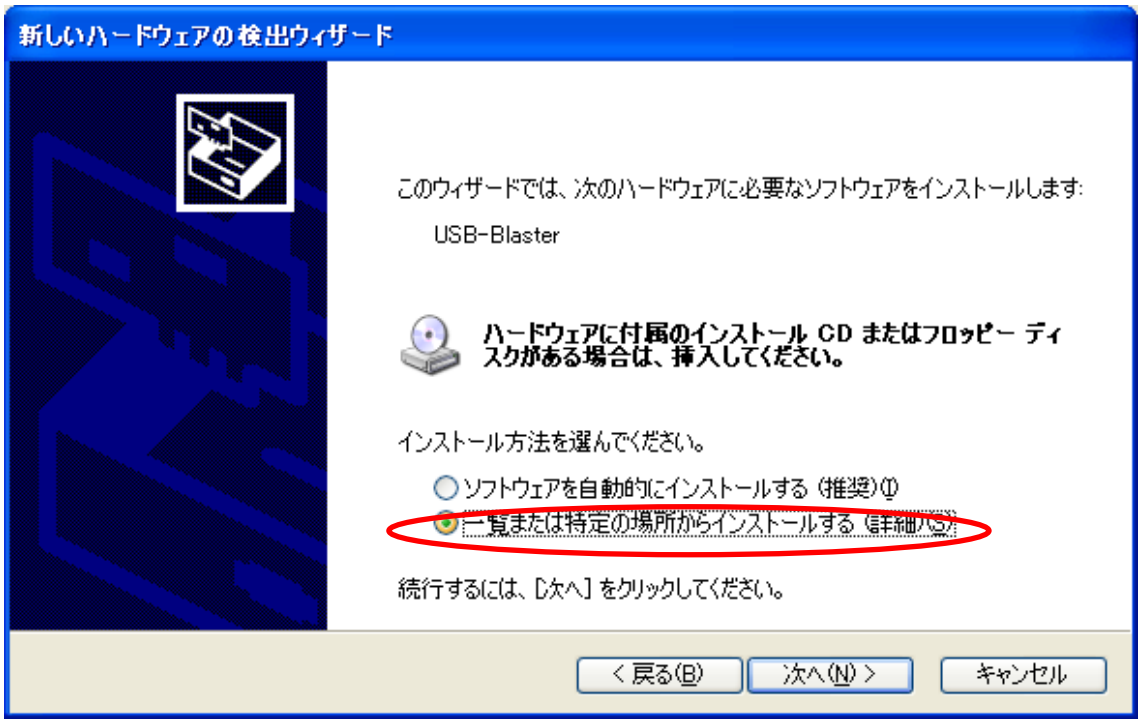
http://www.csun.co.jp/SHOP/200901025.html

次に示す手順に従って、USB-Blaster のデバイス・ドライバをインストールしてください。

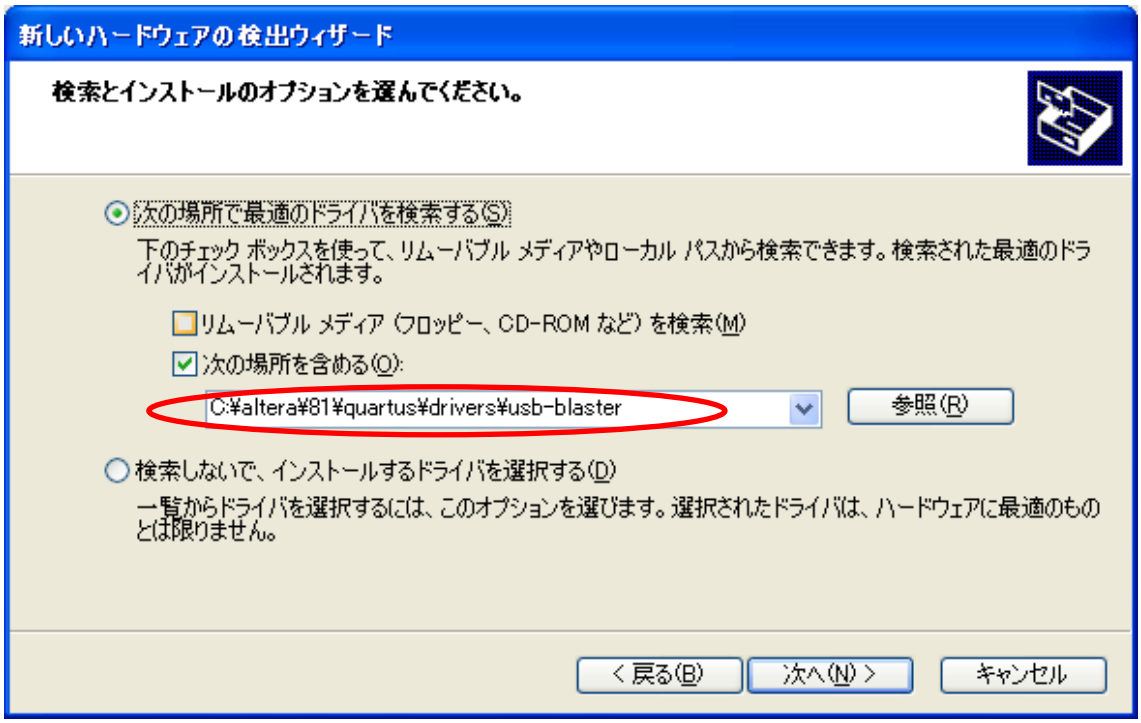


USB-Blaster を USB ケーブルでパソコンと繋ぐと、自動的にこの画面が現れ、「**いいえ、今回は接続しません**」を選択してください。

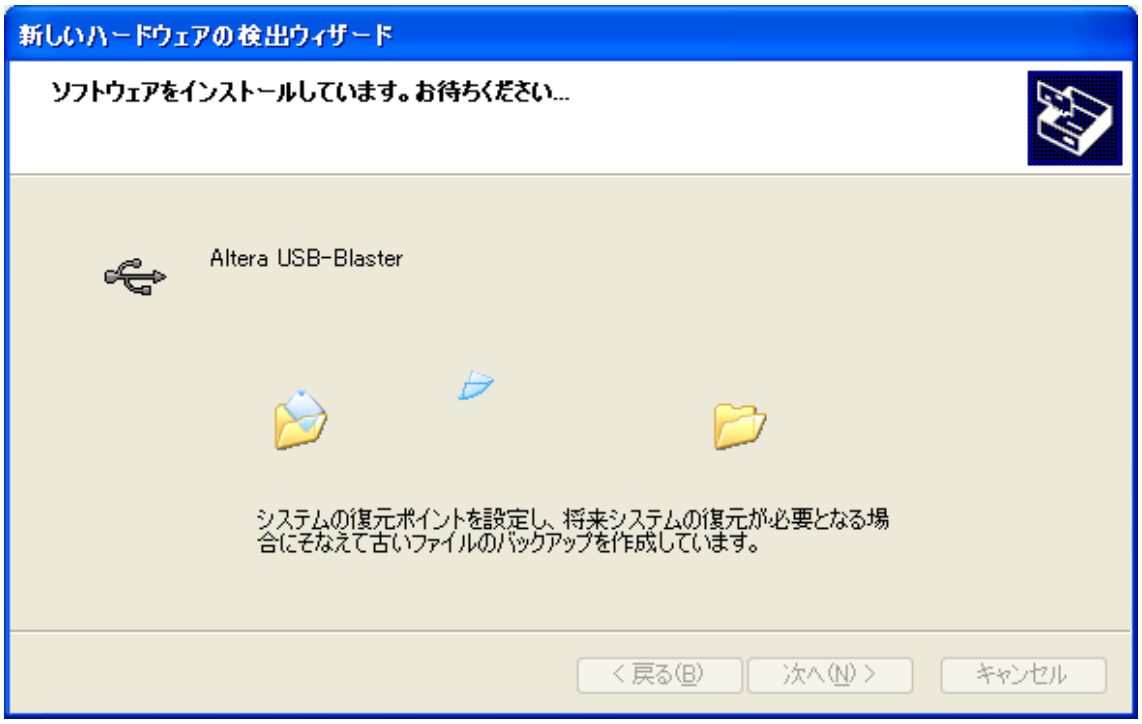




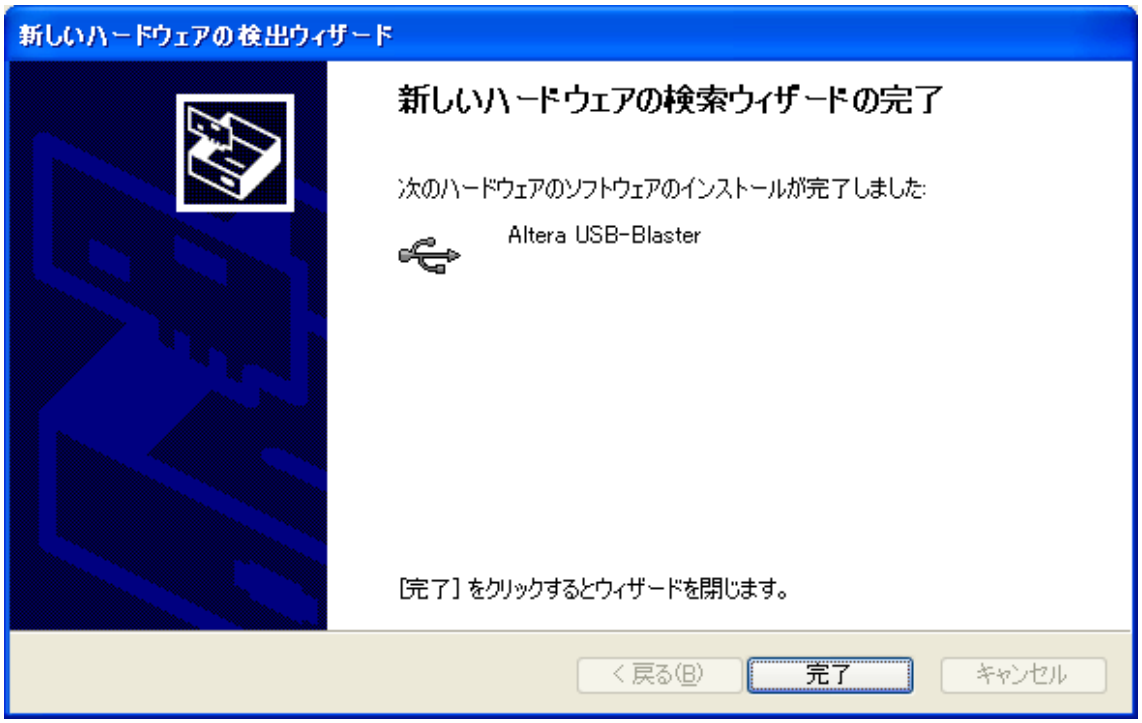
「一覧または特定の場所からインストール」を選択してください。



USB-Blaster のドライバは C:¥altera¥81¥quartus¥drivers¥usb-blaster にあります。



インストール中。

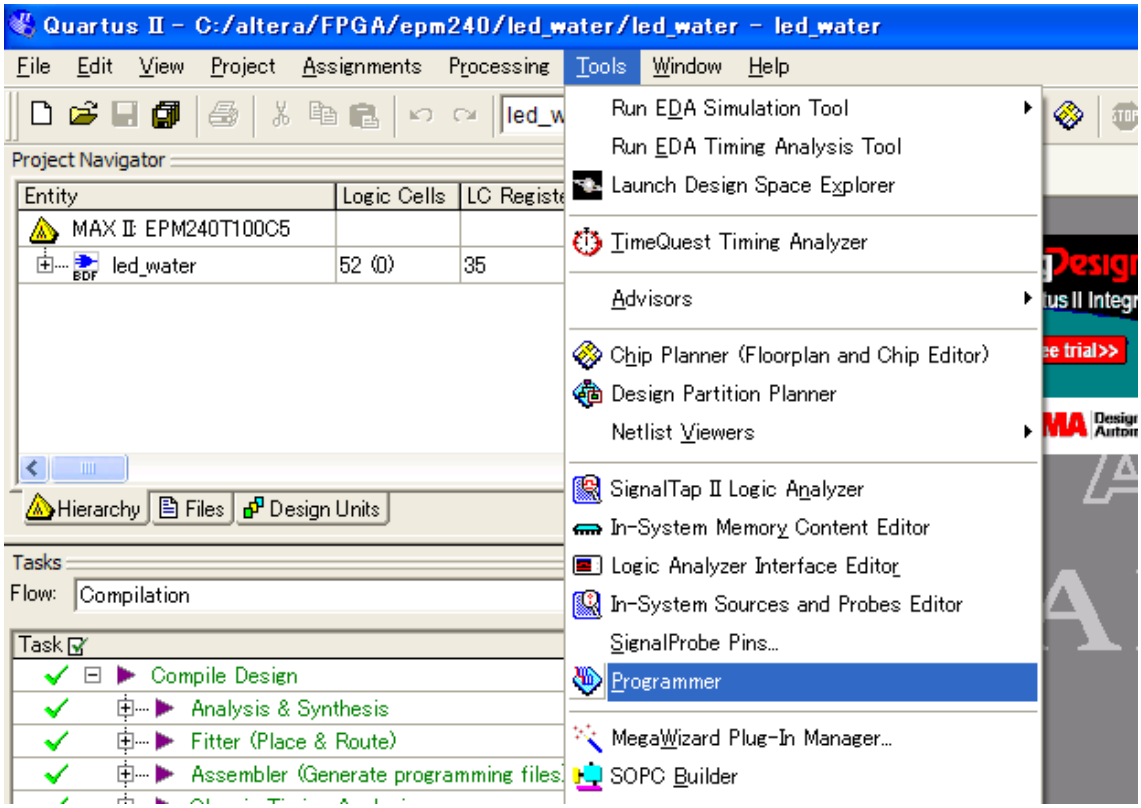


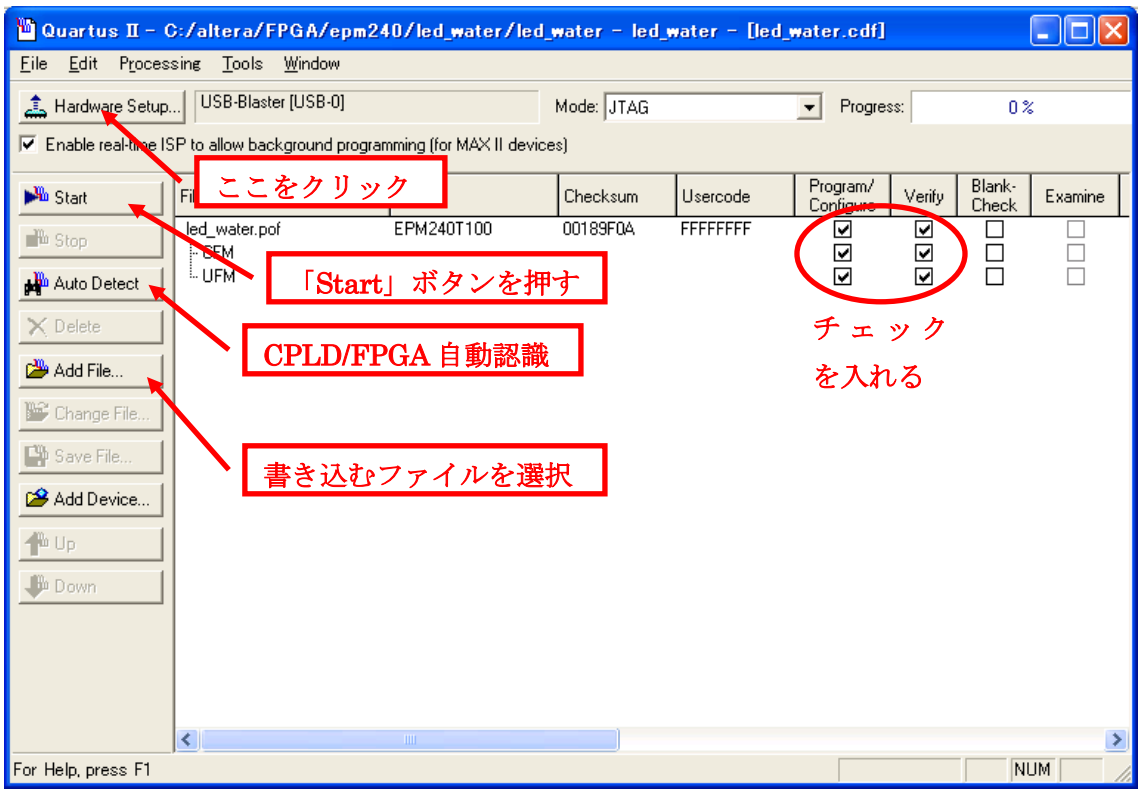
インストール完了します。

3.3 書き込むソフトウェアを起動する

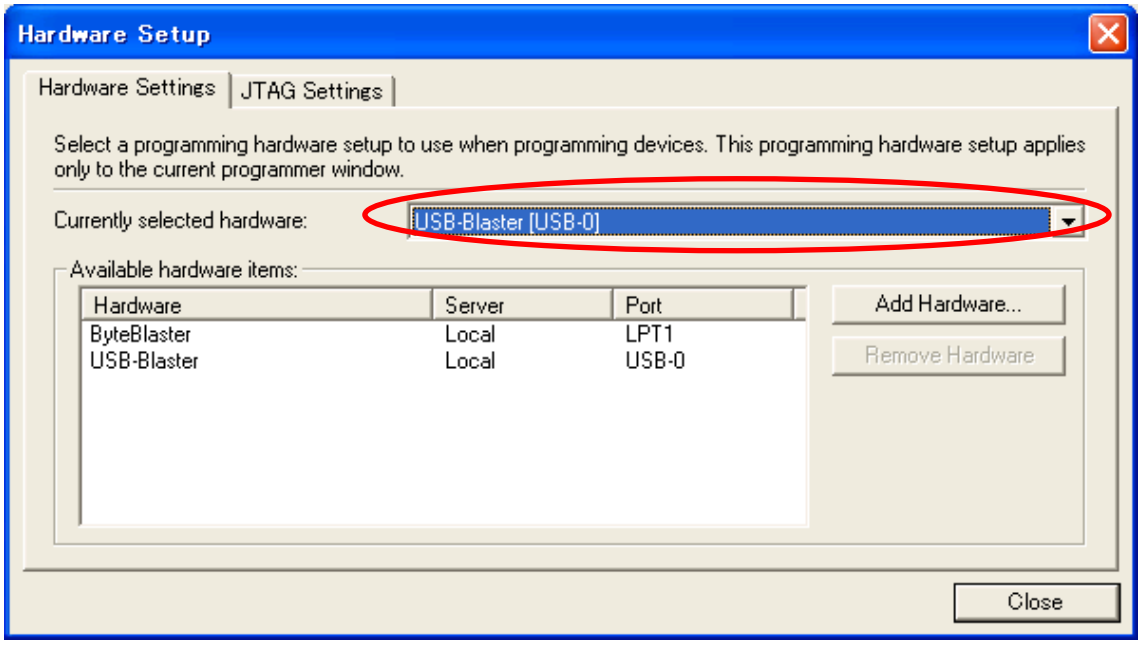
Quartus II の「Tools」メニューから「Programmer」を選択すると、MAX II/Cyclone II

に回路を書き込むソフトウェア「Programmer ツール」が起動します。





Programmer ツールが起動したら、最初に書き込みケーブルのセットアップを行います。「Hardware Setup」というボタンを押してください。



「USB-Blaster[USB-0]」を選択します。「Close」を押して、Hardware Setup ダイアログを閉じたら、「Auto Detect」というボタンを押してください。これは、ケーブルの先にある CPLD/FPGA を自動認識する操作です。うまく CPLD/FPGA が認識されると、EPM240 又は EP2C5 又は EP2C8 という CPLD/FPGA が発見されるはずです。発見されない場合は、

- ケーブルが正しく接続されているか、
- FPGA の場合は、ケーブルとボードの JTAG ポートを繋ぎますか
- CPLD/FPGA 基板に電源が入っているか

など、これまでの作業に問題がないか再度チェックをしてください。

CPLD/FPGA の認識に成功すると、「Add File」ボタンを押して、書き込みファイルを添加します。*.pof は CPLD 用書き込みファイル、*.sof は FPGA 用書き込みファイルです。*.pof の右側にある Program/Configure と Verify の欄にチェックを入れて、「Start」ボタンを押します。プログレス・バーが 100%まで達すれば、書き込み成功です。

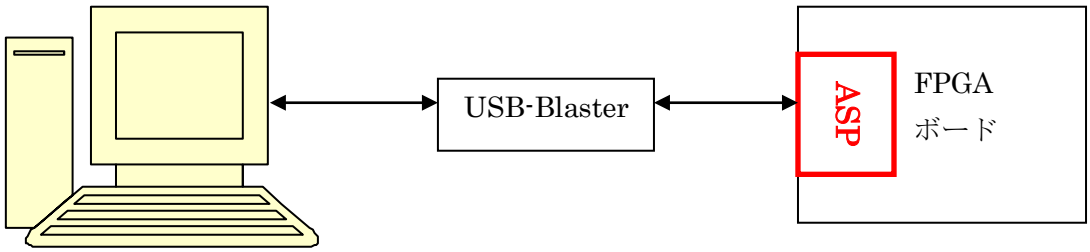
CPLD/FPGA 用 I/F 基板上の LED が点滅しているのを確認してください。どうでしょうか？うまく点滅したでしょうか。

3.4 FPGA のコンフィギュレーションデバイスに書き込む

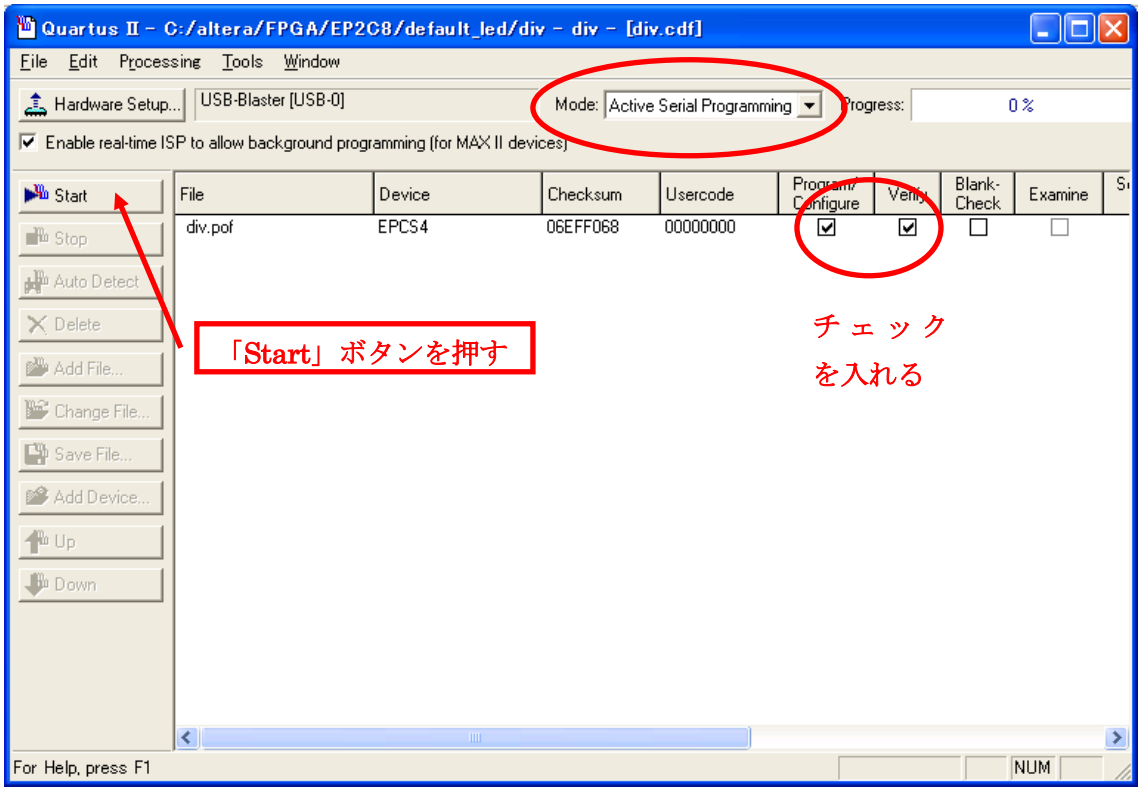
Cyclone II は SRAM ベースの FPGA なので、電源投入直後は中身が空の状態です。製品化

の際や、電源投入後に自動的に動作させる必要がある場合は、専用のコンフィギュレーションデバイス(EPCS4)に回路情報を書き込む必要です。

専用のコンフィギュレーションデバイスに書き込む手順：



先ず、USB-Blaster と FPGA ボードの ASP ポートを繋ぎます。
書き込むソフトウェア「Programmer ツール」が起動します。



「Mode」に[Active Serial Programming]を選択します。「Add File」ボタンを押して、書き込みファイル***.pof** を添加します。*.pof の右側にある Program/Configure と Verify の欄にチェックを入れて、「Start」ボタンを押します。プログレス・バーが 100%まで達すれば、書き込み成功です。

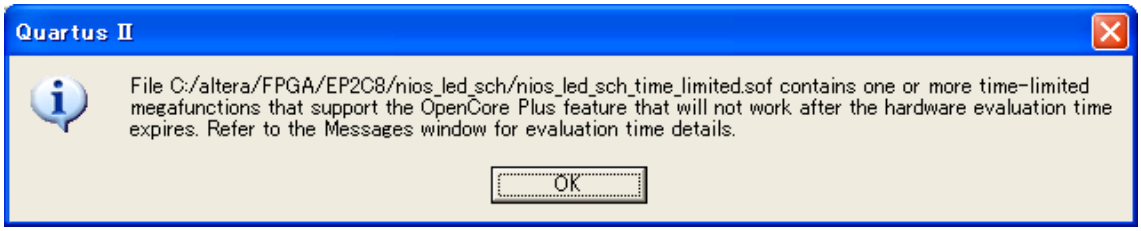
書き込み成功した後、USB-Blaster を FPGA ボードの ASP ポートから抜いて、FPGA ボードに電源を再投入すると、どうの現象が出てきますか？

3.5 NIOS II プロセッサーの初体験

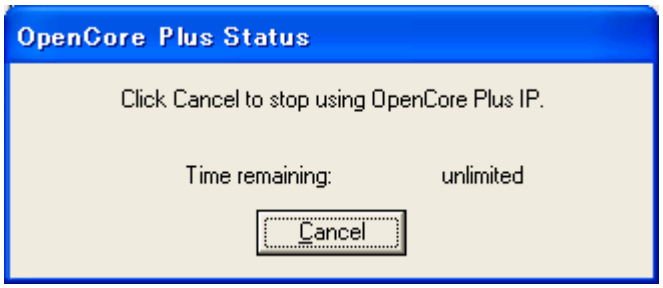
エクスプローラまたはマイ コンピュータを起動して、
C:¥altera¥EP2C8¥_1_key_led_test
というフォルダを開いてください。

これらの中に、名前が nios_test.qpf、Quartus II Project File となっているファイルがあります。これをダブル・クリックすると、Quartus II が起動して、nios_test.qpf というプロジェクトが開きます。

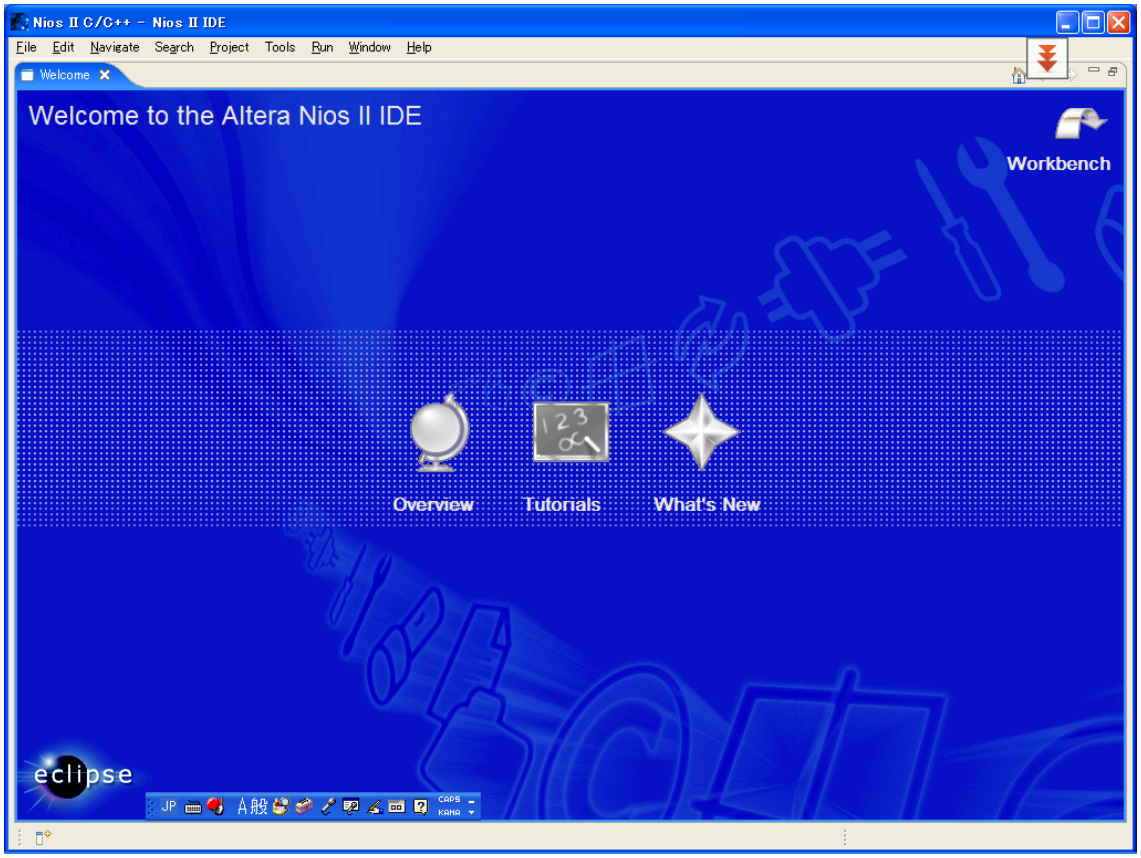
他のプロジェクトと同じ手順でコンパイルして、Cyclone II ボードに書き込みます。
書き込み時、このような情報が出てきます。正式製品なら、アルテラ社からライセンスが必要です。評価の場合は、そのまま「OK」ボタンを押します。



書き込み完了したら、その画面が出てきます。「Cancel」ボタンを押さないでください。その画面をそのまま置いといてください。

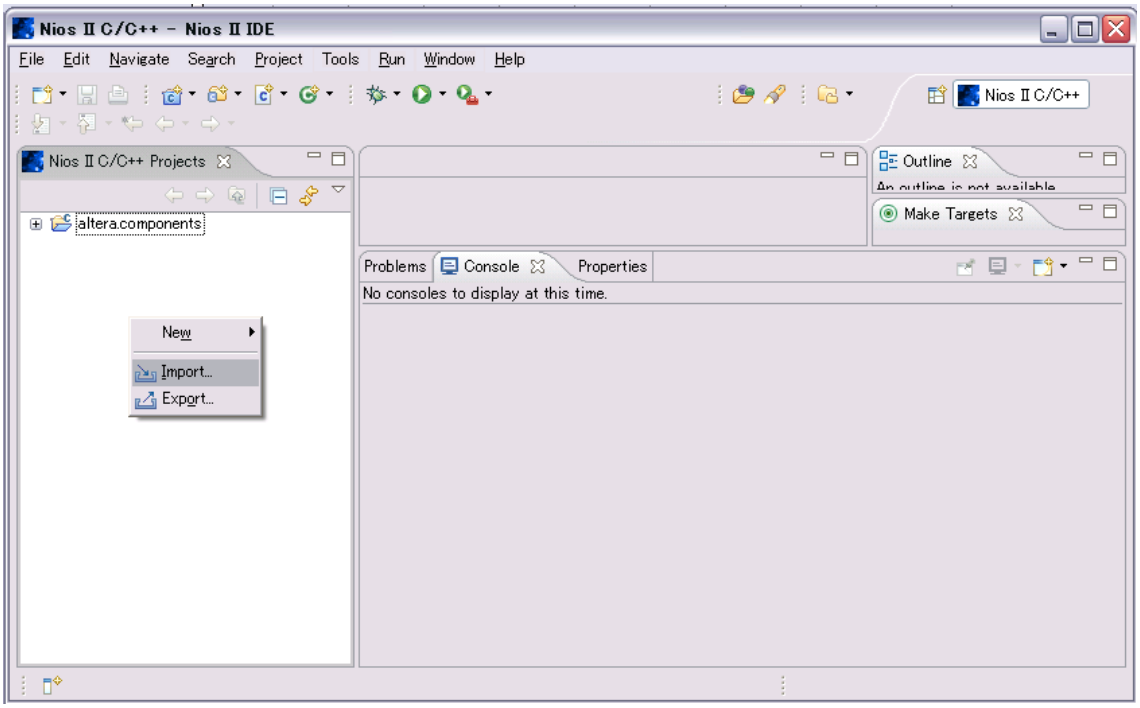


Windows の「スタート」→「すべてのプログラム」→「Altera」→「NIOS II EDS 8.1」から NIOS II 8.1 IDE が起動します。

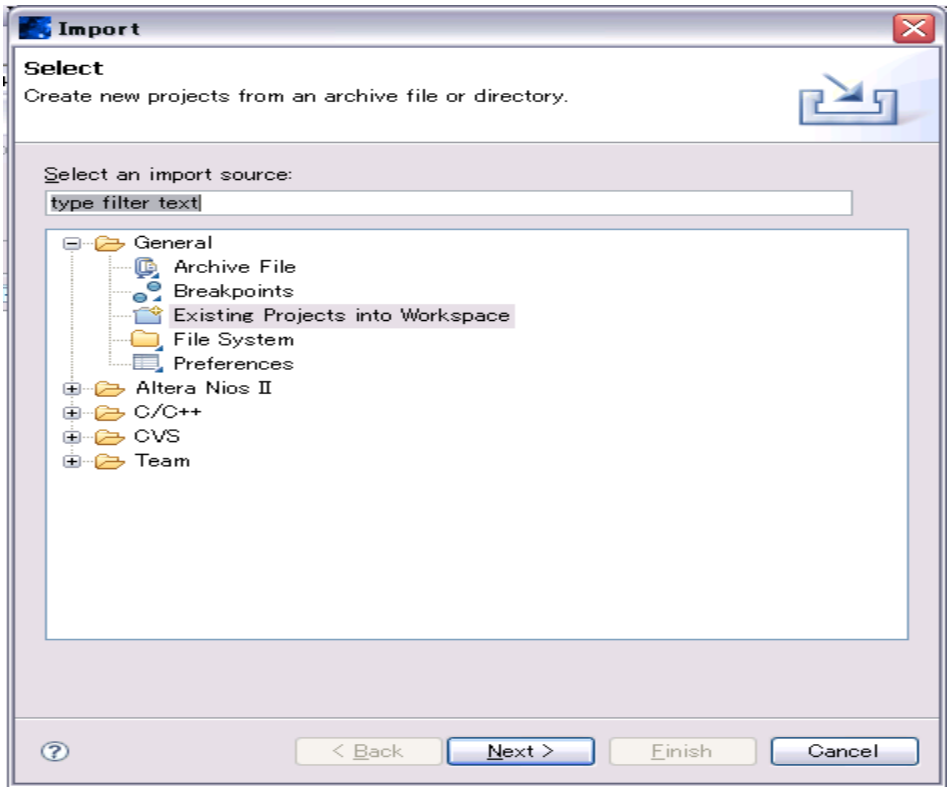


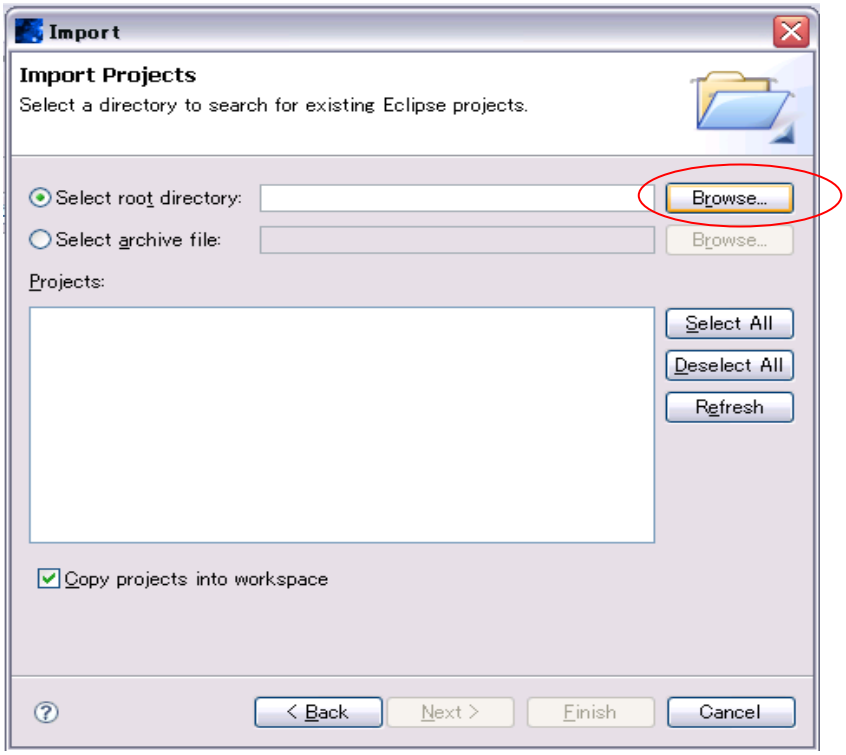
NIOS II IDE の初起動の画面です。

NIOS II IDE の project window で右クリックし Import を選択します。

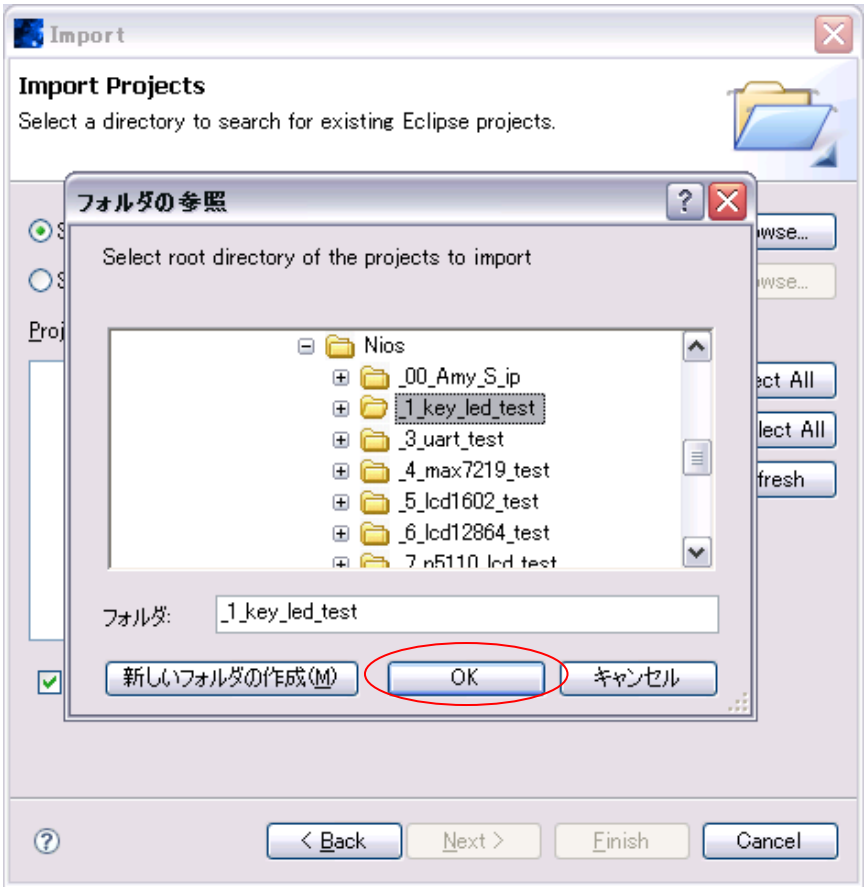


「Import」画面で「Existing Projects into Workspace」を選択して、「Next」ボタンを押します。

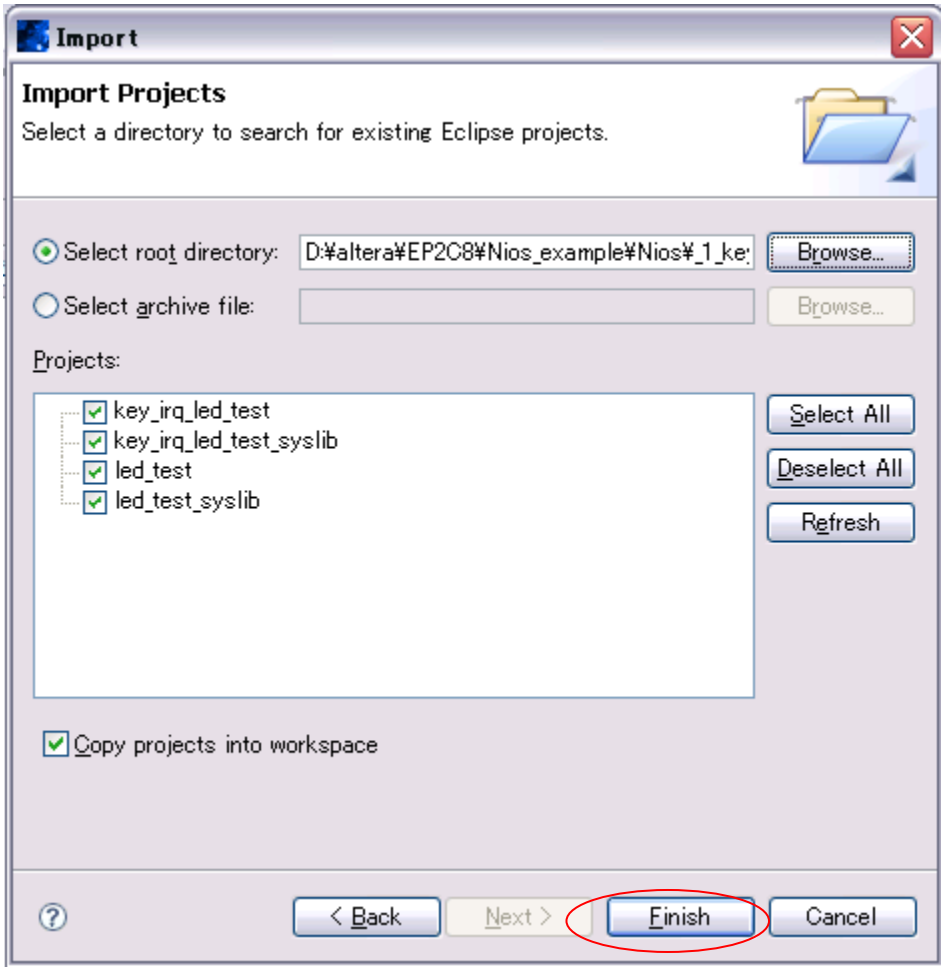




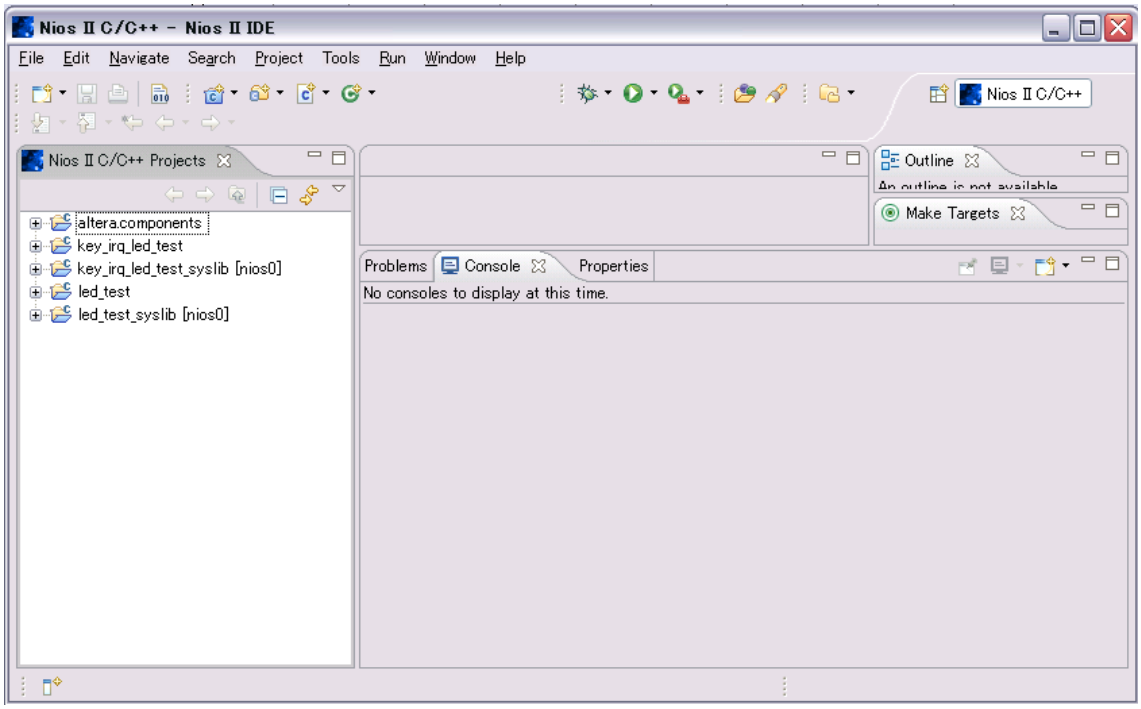
「Browse」ボタンを押してプロジェクト保存しているフォルダを選択します。



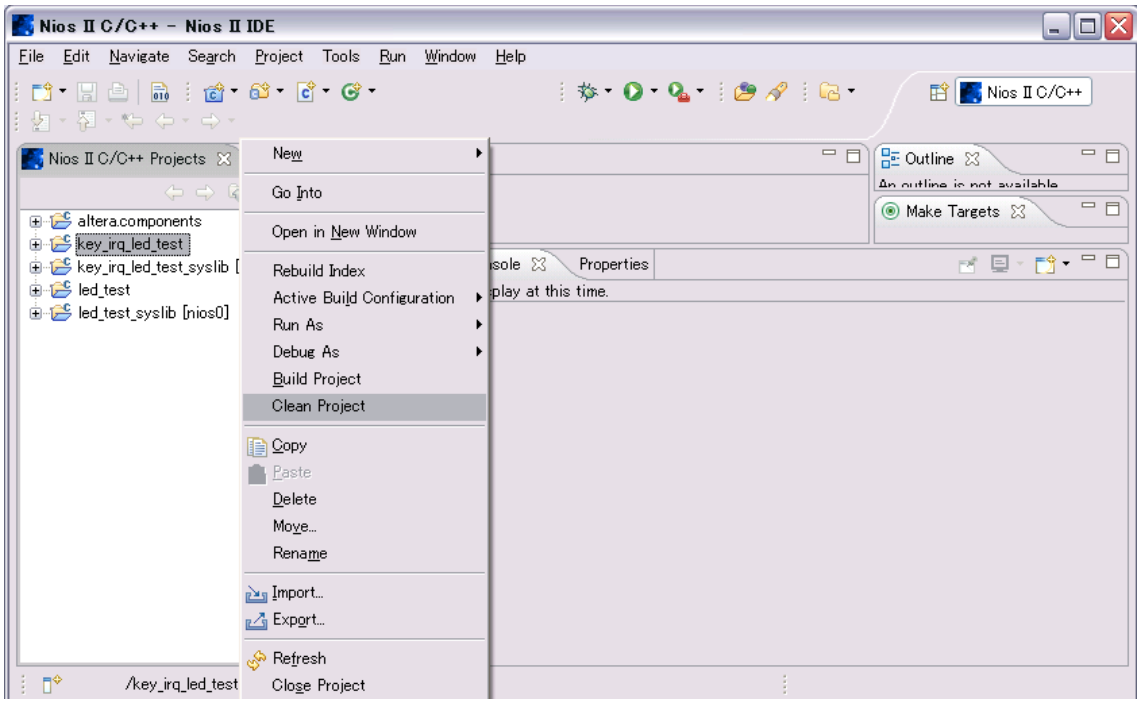
「OK」ボタンを押します。



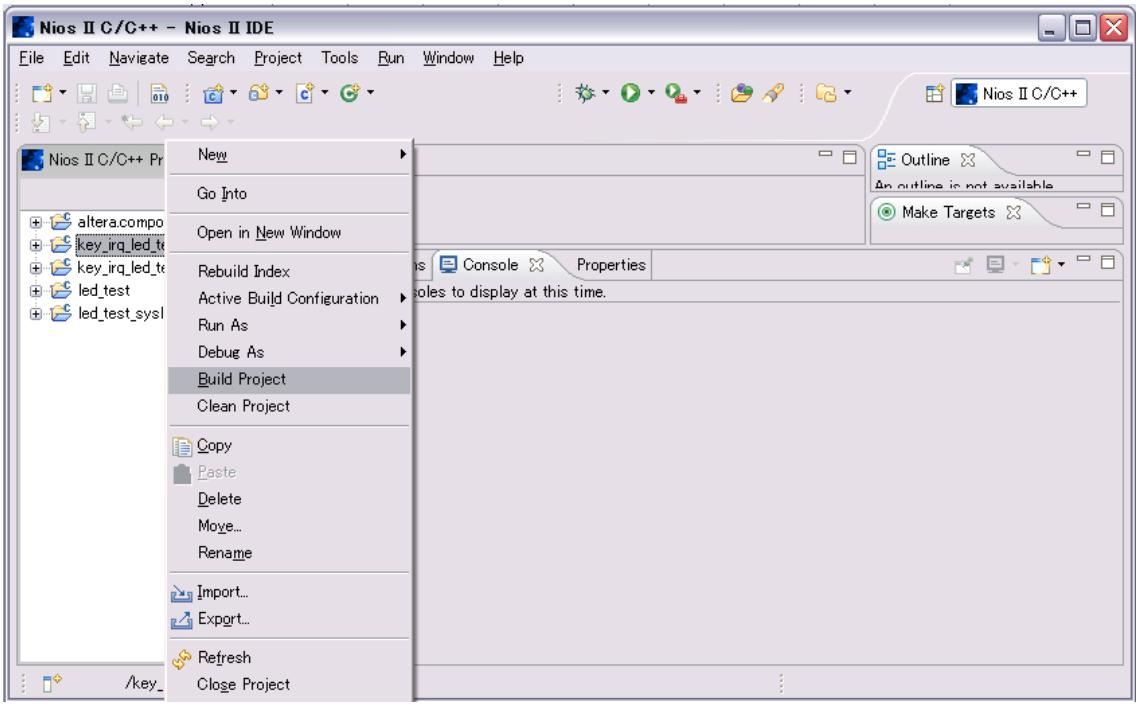
そのまま、「Finish」ボタンを押します。



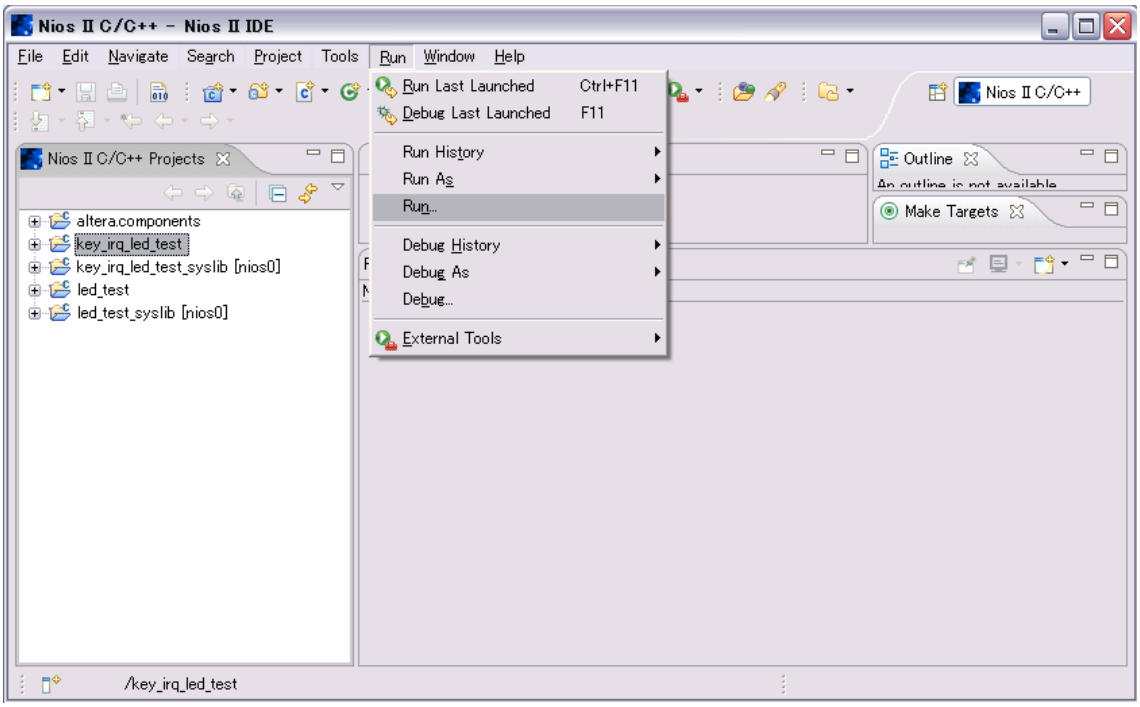
このままビルドすると旨く行かない場合がありますので、一回 Clean Project を実行します。

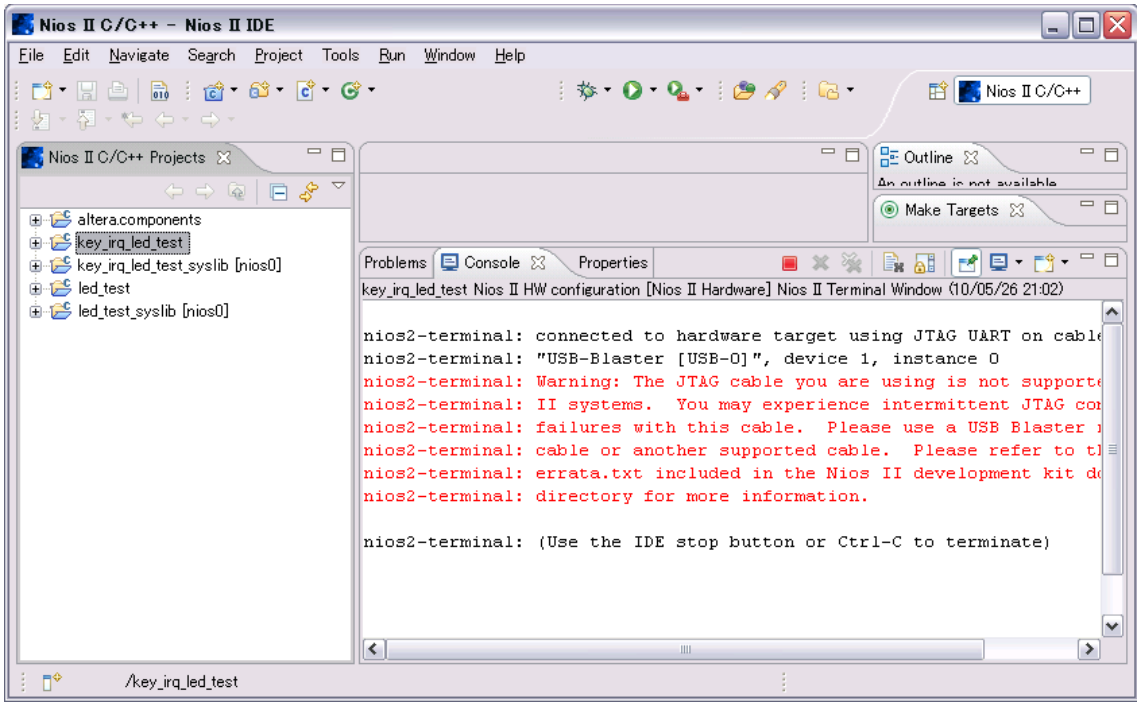


左側の「key_irq_led_test」でマウスの右ボタンをクリックして、「Build Project」を選択して、ビルドを開始します。



終わりましたら、NIOS II IDE のメニュー「Run」→「Run」を選択します。





上記コンソール内赤文字部分出てもプログラムの実行には影響ないです。

実行結果としては、実験 I/F ボードのキーより LED 点灯・消灯します。

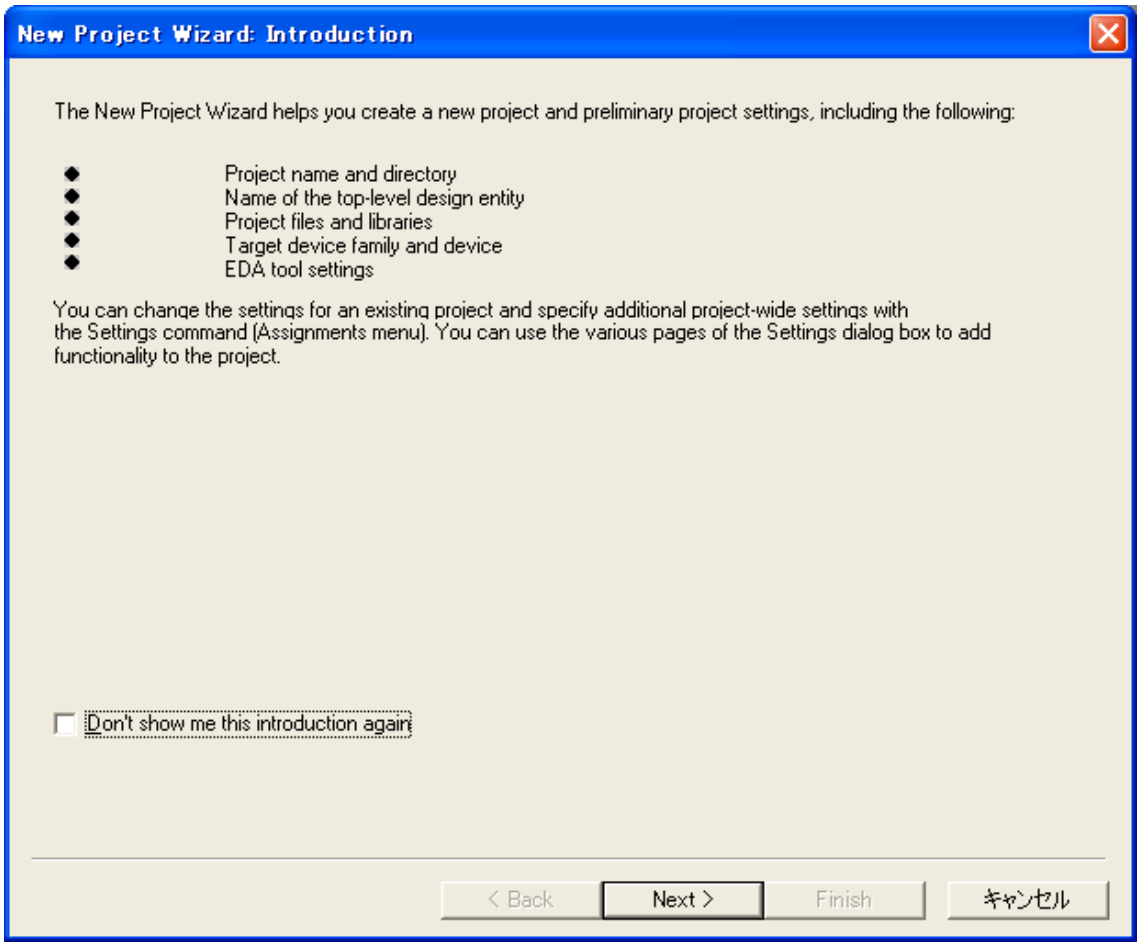
第四章 CPLD/FPGA の開発入門

4.1 プロジェクトを作成する

Windows を起動し、「スタート」メニューから Quartus II Web Edition を起動します。

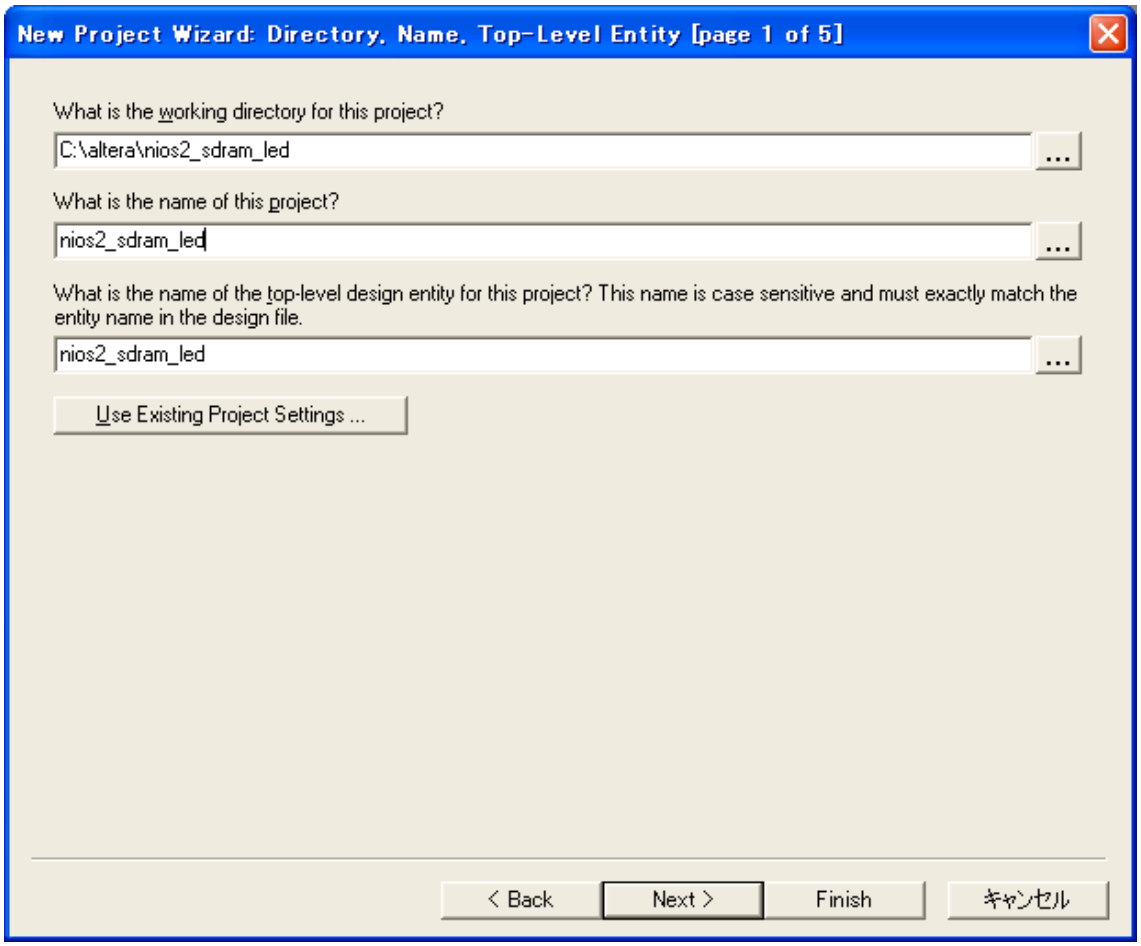
Quartus II 評価版上では、これから作る回路が一つのプロジェクトとして使扱われます。まずは、新しいプロジェクトを作成しましょう。

新規にプロジェクトを作成するには、Quartus II 評価版の「File」メニューから「New Project Wizard」を選択し、プロジェクト作成ウィザードを起動します。このウィザードを使えば、ダイアログに表示された質問に答えていくだけで、簡単にプロジェクトを作ることができます。



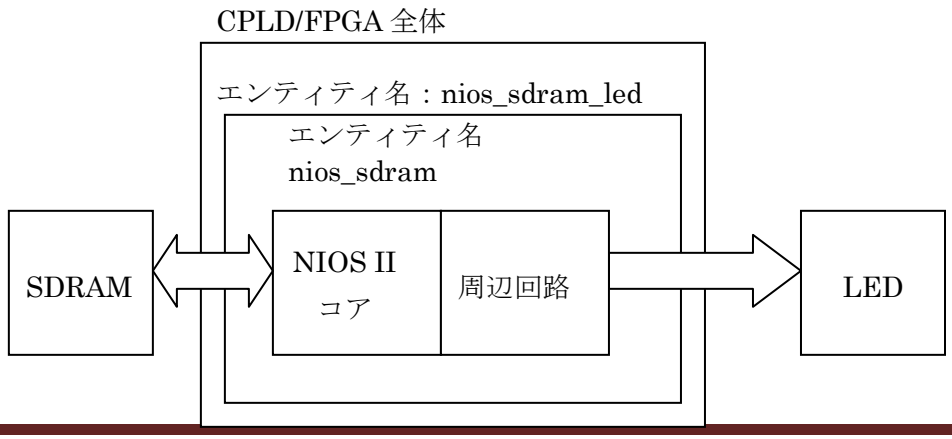
最初に、New Project Wizard に関する説明が表示されるので、そのまま「Next」を押します。すると、プロジェクトの名前や保存場所を聞いてきます。各問いに対して、次のよう

に書き入れて「OK」ボタンを押してください。



パス名やプロジェクト名、モジュール名に漢字や空白などの特殊な記号が含まれていると、ツールによってはうまく動かないものがあるので、半角のアルファベットや数字などを組み合わせただけの単純な名前にしてください。

プロジェクト作成ウィザードで入力した三つめの項目は、回路を階層的に設計する場合に最上位の階層に置くエンティティの名前です。エンティティとは、あるまとまった機能をもった回路のことです。

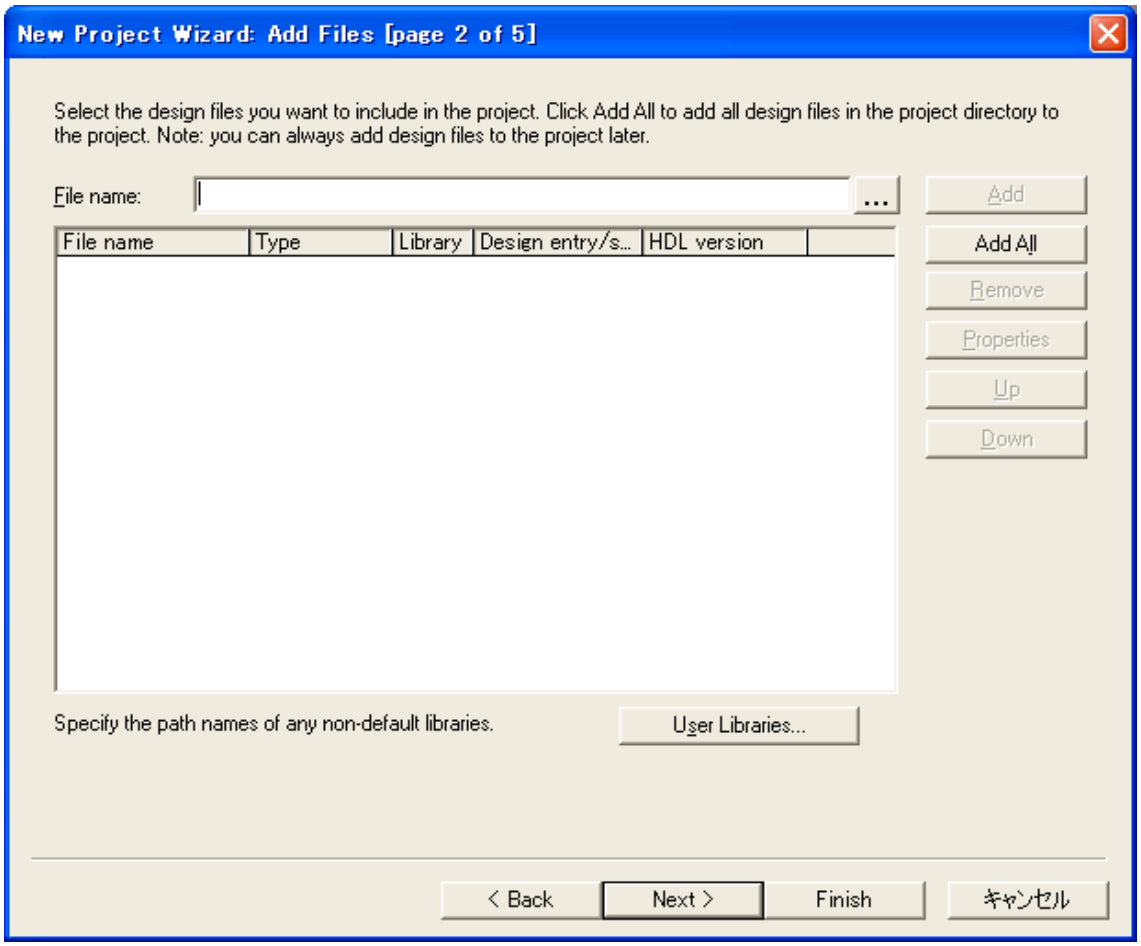


これから作成する回路のエンティティの階層構造です。FPGA 内部は、nios_sdram_led というエンティティで作られています。そして、nios_sdram_led というエンティティの中に、nios_sdram というエンティティがあります。さらに、nios_sdram というエンティティの中に NIOS II コアと周辺回路など幾つのエンティティがあります。

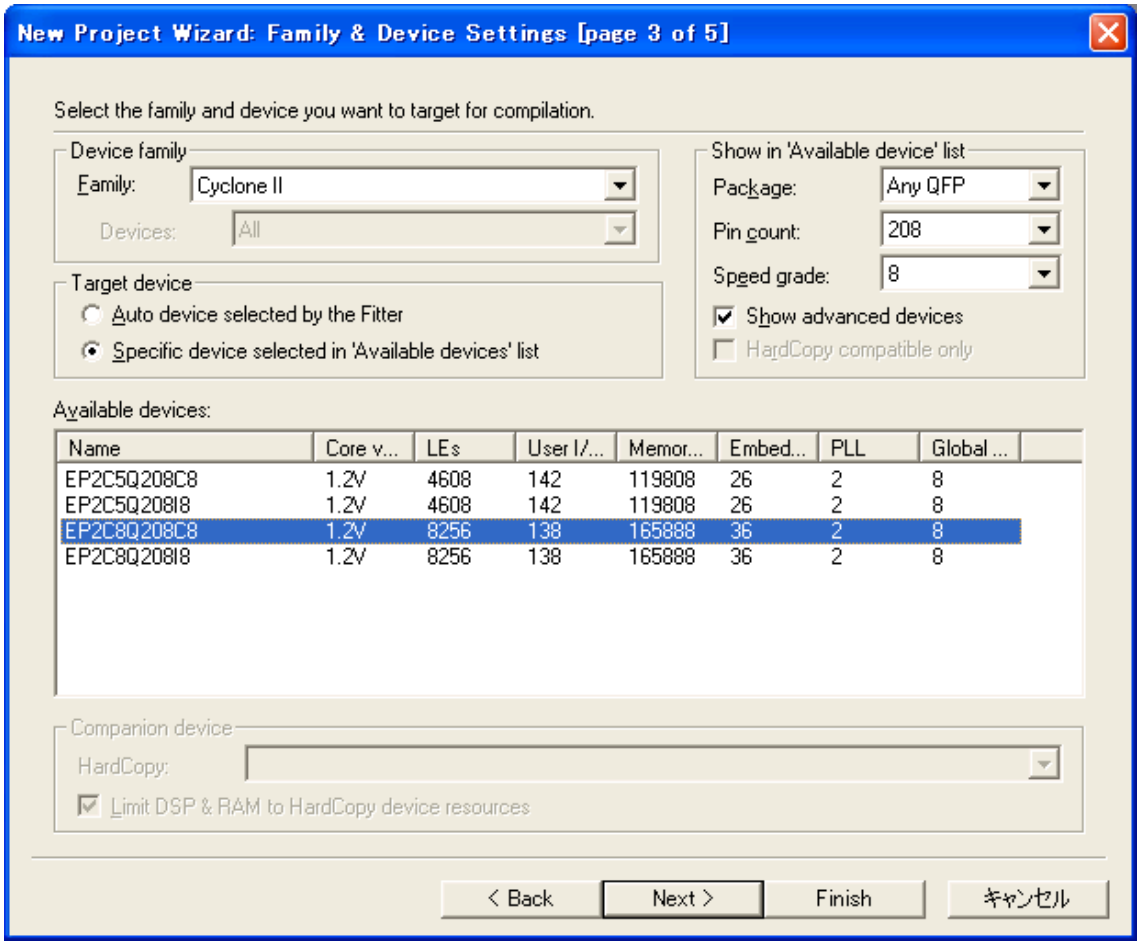
※ MAXII は容量が不足ですので、ソフトプロセッサ NIOS II を搭載できません。

このように、小さなエンティティを組み合わせて大きな回路を作って階層構造にすると、効率良く開発できます。

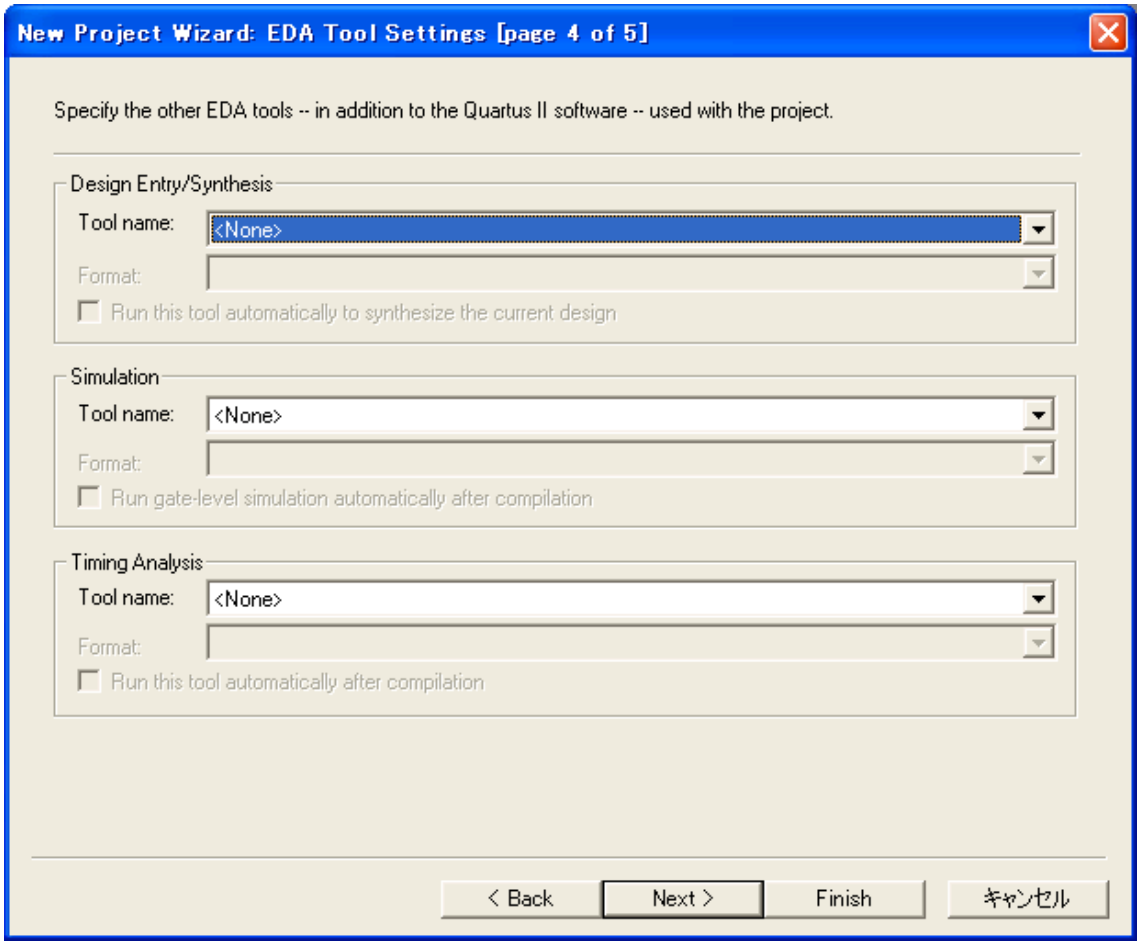
ダイアログの三つの欄に入力したら「Next」を押します。すると、



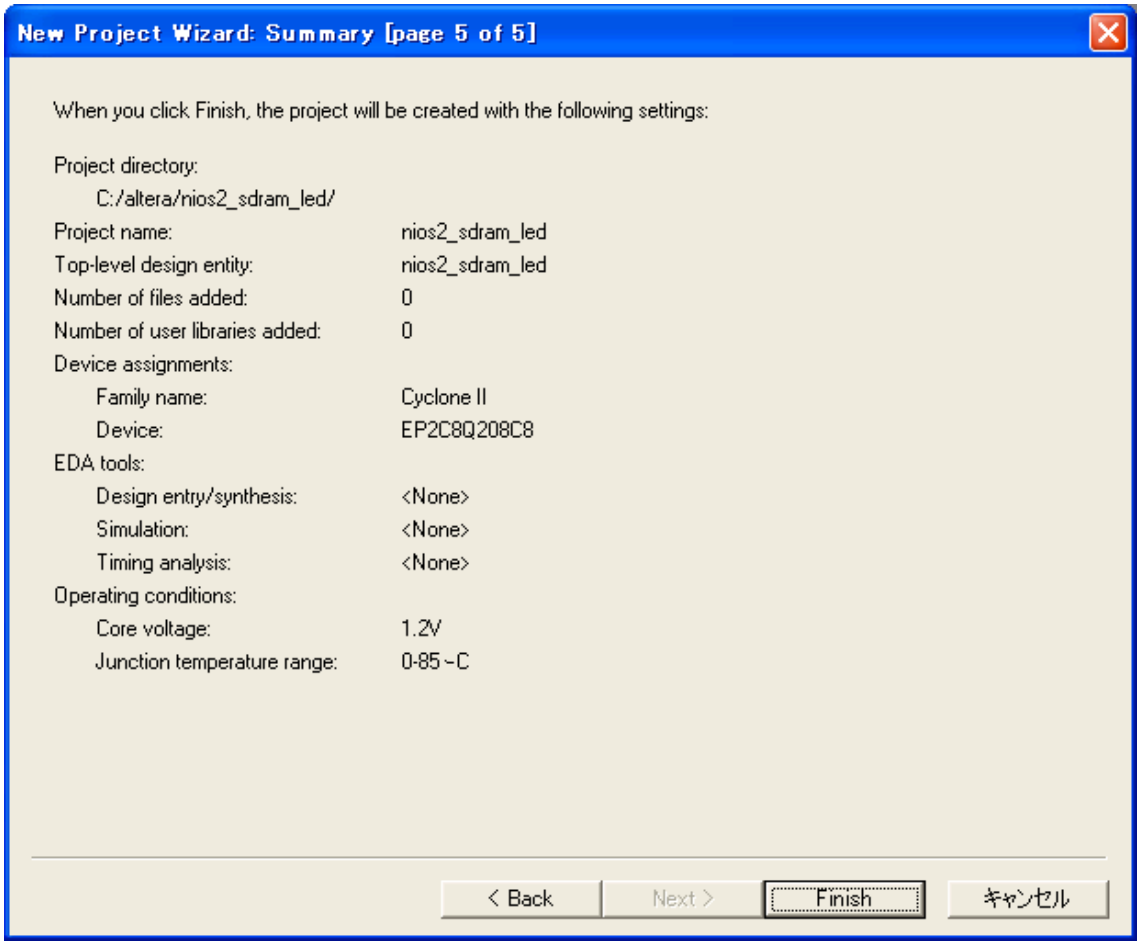
この画面が現れます。プロジェクトに追加したいファイルがある場合はここで追加できます。今回は不要なので、なにも選択せずに「Next」を押します。



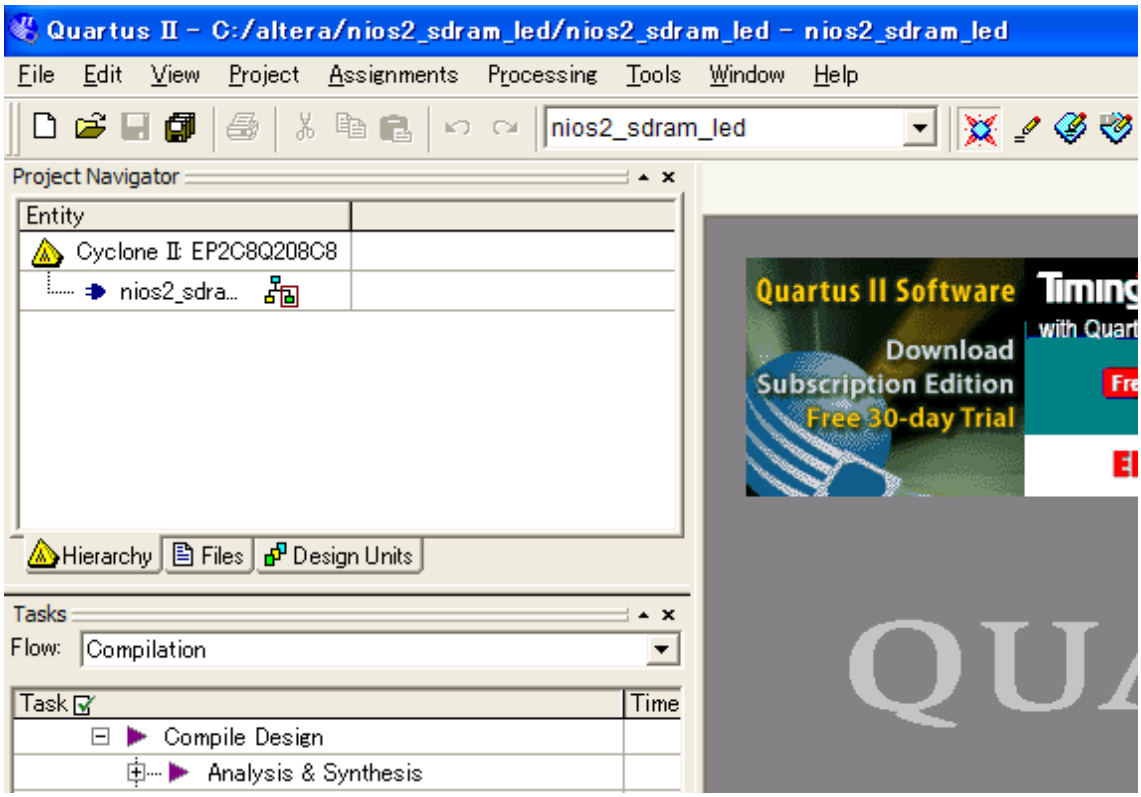
使うFPGAデバイスの選択ダイアログで、まず「Family」というダウン・メニューで「Cyclone II」を選択し、「Available devices」の中から「EP2C8Q208C8」を選びます。選んだら、「Next」を押します。



この画面では、このプロジェクトで使用したい Quartus II 評価版以外の外部ツールを選択できます。今回は Quartus II 評価版だけで最後まで設計を行うので、何も選択せずに「Next」を押します。



内容を確認して「Finish」を押します。



これは nios_sdram_led プロジェクト作成直後の画面です。

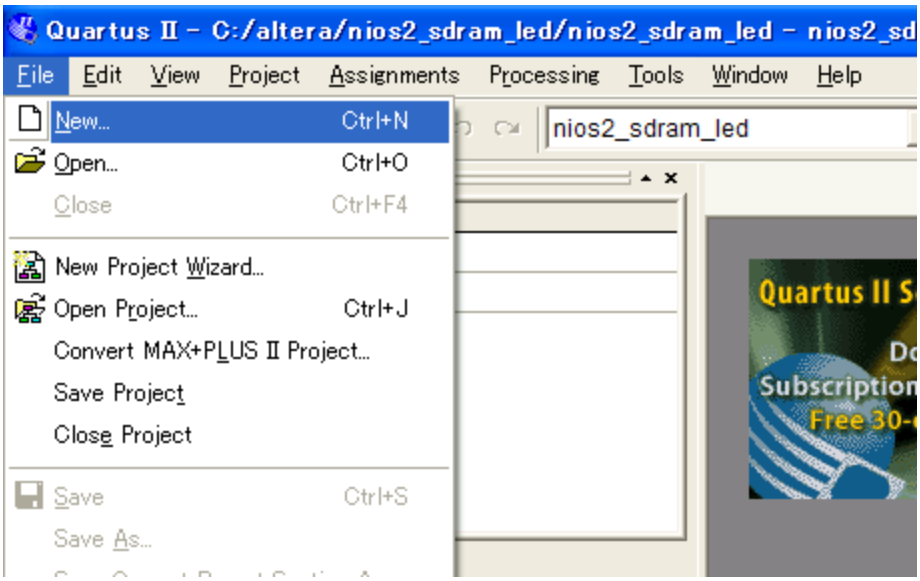
左上の Project Navigator の中に、デバイス名とトップ・レベル・エンティティ名(nios_sdram_led)が入れ子になって表示されています。これは「EP2C8Q208C8 というデバイスの中に nios_sdram_led というエンティティが入っていますよ」という意味です。

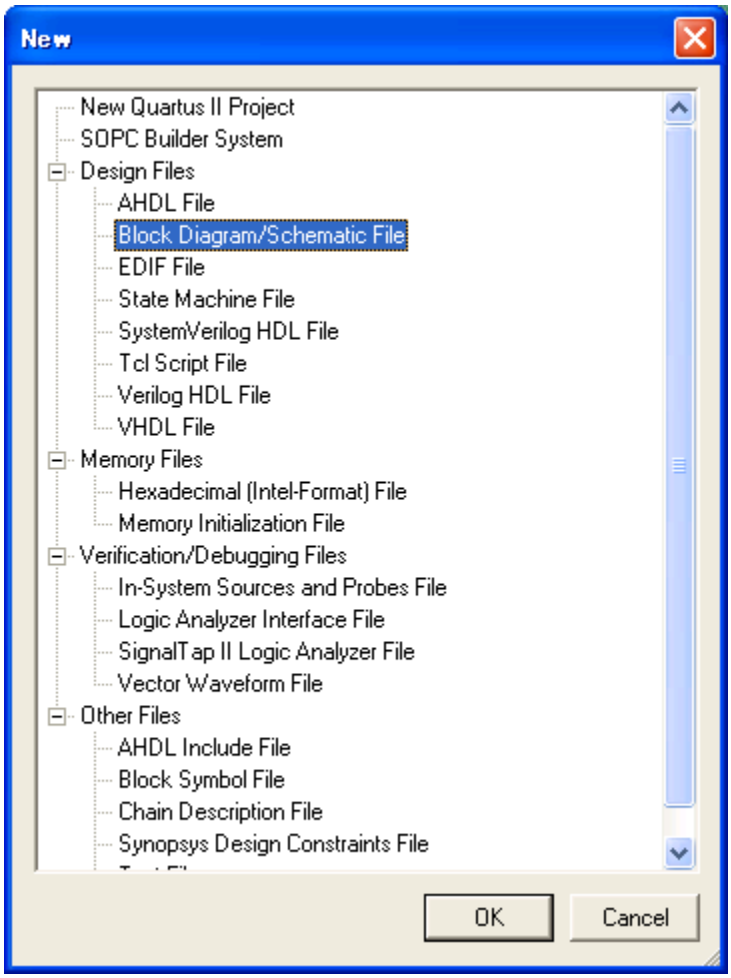
このように、Project Navigator 欄を見れば、今開いているプロジェクトがどのような構造をしているかを確認できます。Project Navigator 欄を閉じてしまった場合は、「View」メニューの中の「Utility Windows」という項目の中に「Project Navigator」があるので、こちらを選択すればもう一度表示できます。

4.2 エディタで回路図を描く

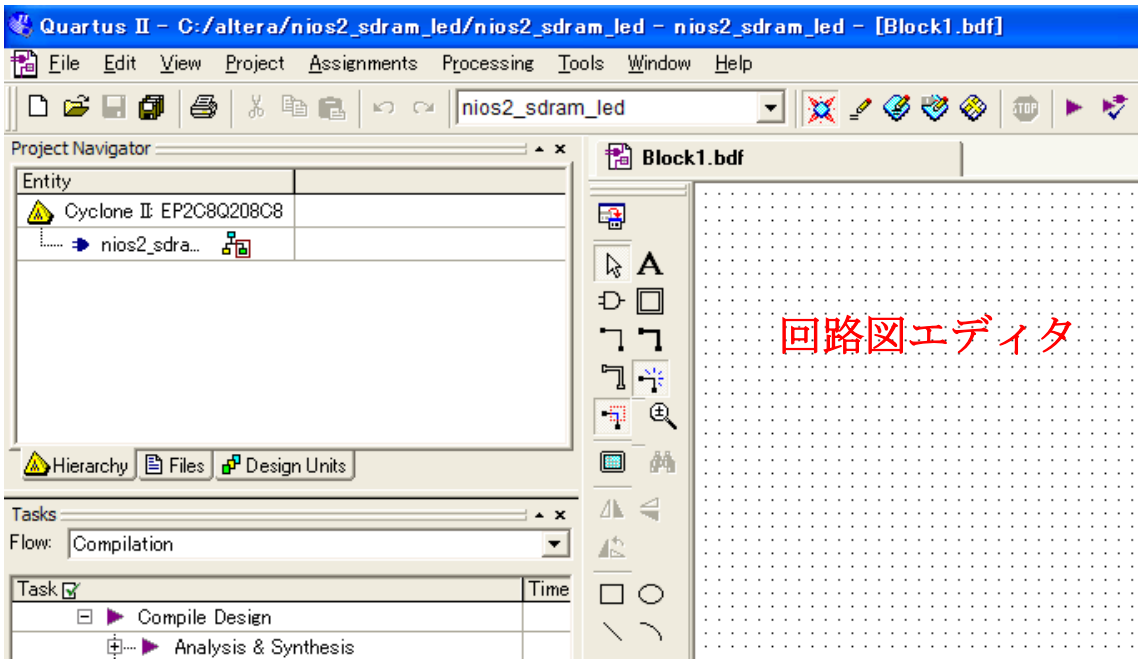
4.2.1 トップ・エンティティを作成する

「File」メニューから「New」を選択します。



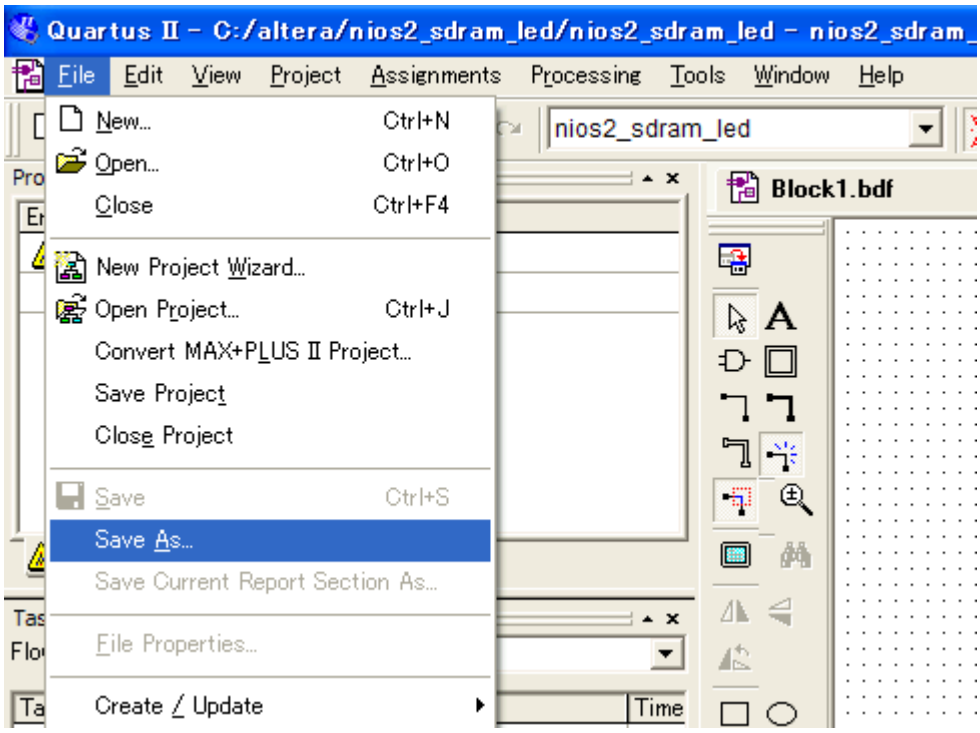


「Block Diagram/Schematic File」を選択します。「OK」を押します。

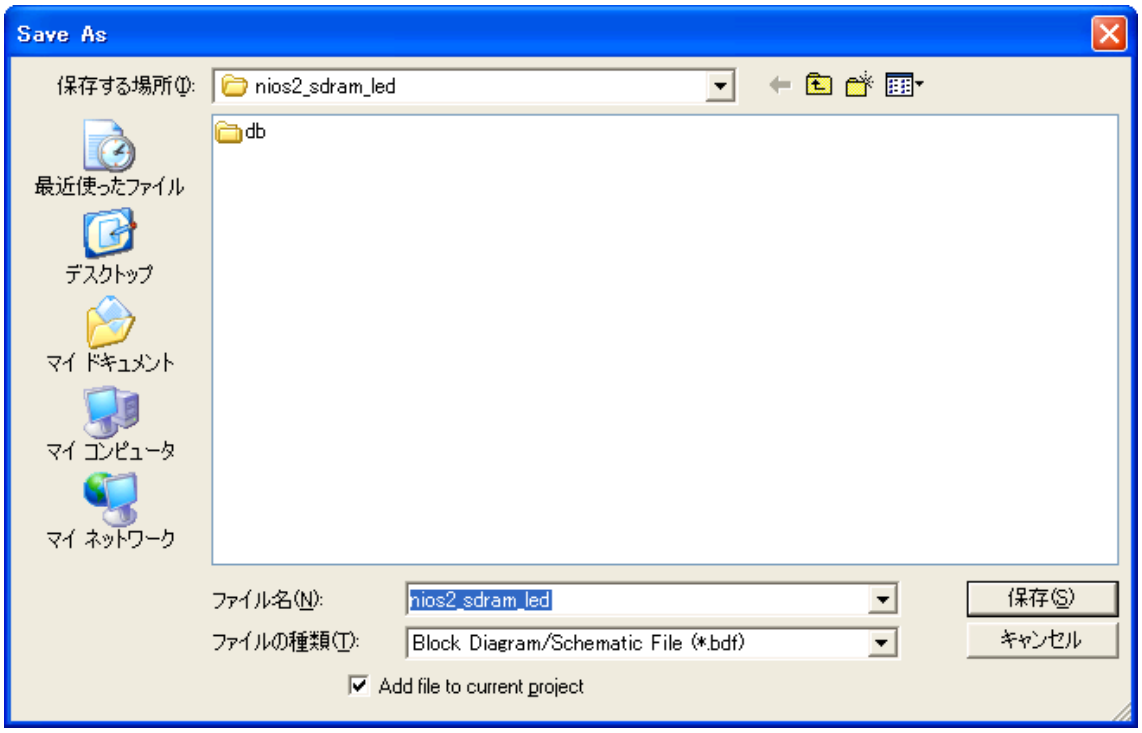


これが回路図エディタです。ここに回路記号を配置していけば、CPLD/FPGA 内部の回路を設計できます。

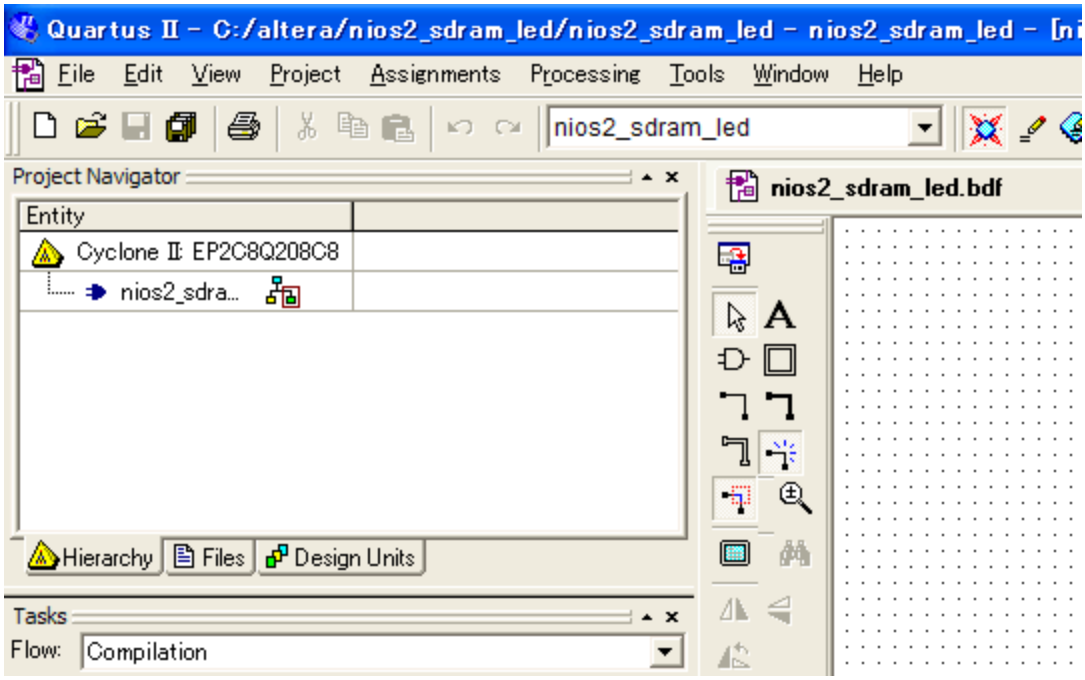
Block1.bdf という名前は、Quartus II 評価版が勝手に付けた名前です。これから作りたいのは nios_sdram_led というエンティティです。回路図とエンティティを対応させるために、ファイル名を nios_sdram_led.bdf として保存します。



「File」メニューから「Save As…」を選び、ファイル名を nios_sdram_led としてください。保存する場所は先ほど指定したプロジェクトのワーキング・ディレクトリで、
C:¥altera¥nios_sdram_led
です。「Save As…」を選ぶと自動的に、このフォルダが開かれます。



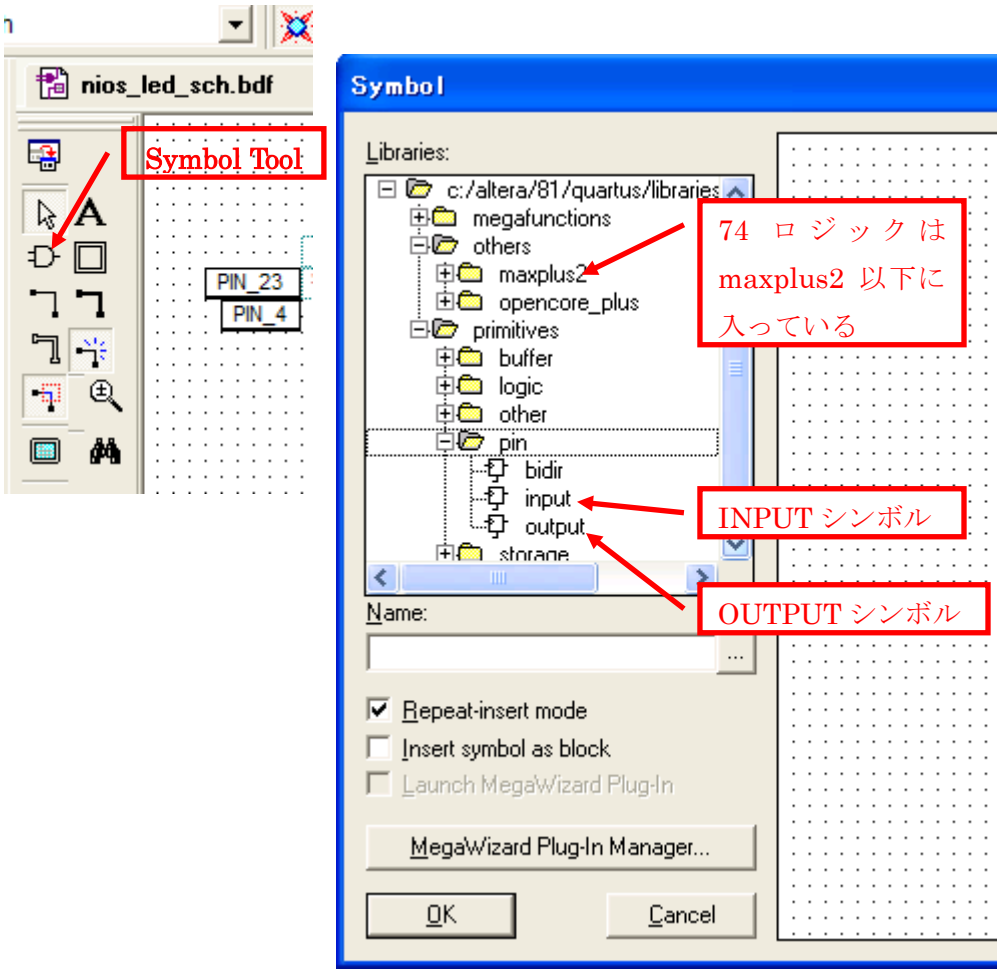
保存が終わると、ウィンドウの名前が nios_sdram_led.bdf に変わります。



4.2.2 作画手順

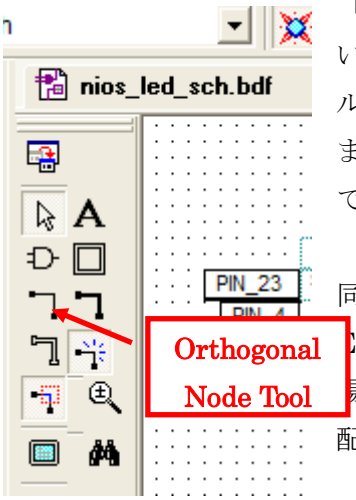
①シンボルを置く

「Symbol Tool」というボタンを押すと、このようなダイアログが開きます。



Libraries 欄に、[+]記号をクリックすると、中身が表示されます。中身のシンボルを選択して「OK」ボタンを押します。マウス・カーソルに選択されたシンボルがくっついた状態になるので、シンボルを配置したい場所に移動して、マウスの左クリックを押します。配置できた点で ESC キーを押してカーソルを元に戻してください。

②シンボル同士を接続する



「Symbol Tool」の少し下にある「Orthogonal Node Tool」という鍵状のアイコンを選択します。マウス・カーソルをシンボルのピンに併せて左ボタンを押し、そのまま離さずに他のピンまでドラッグします。すると、ドラッグの開始点と終了点が線で結ばれます。

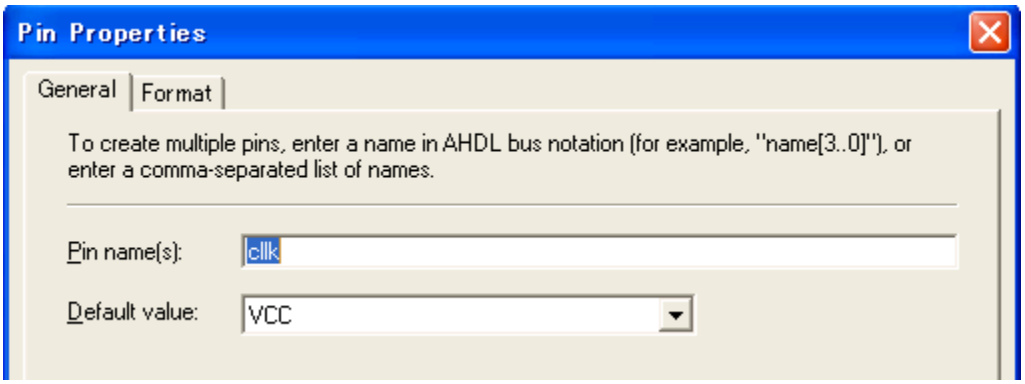
同様にしてほかの線も配置してください。接続が終わったら ESC キーを押し、元のカーソルに戻します。接続を間違えた場合は、一度 ESC キーを押して元のカーソルに戻し、不要な配線をクリックして Delete してください。

③INPUT 端子と OUTPUT 端子の名前を変える

INPUT シンボルと OUTPUT シンボルは、エンティティ外側と接続するための端子です。端子の名前は pin_name という素っ気ないものになっています。

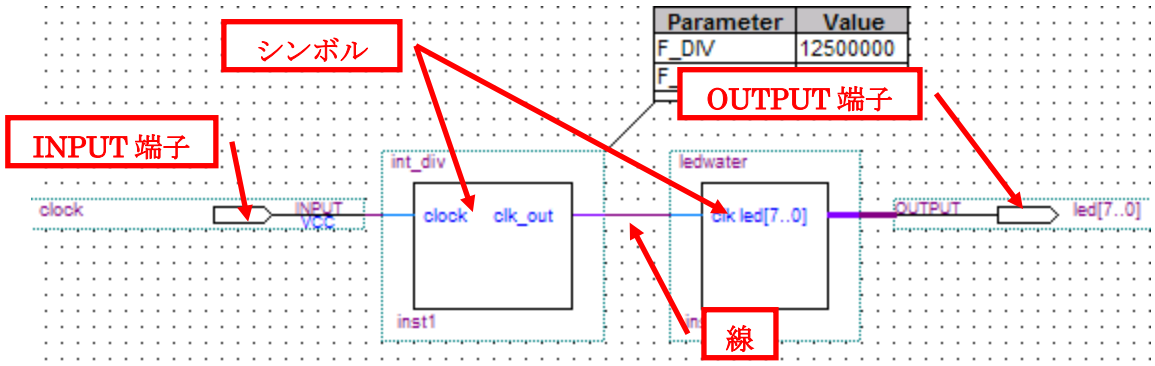
これでは、エンティティを使うときに、いったい何のための端子なのかわからなくなります。そこで、わかりやすい名前に変更しておきます。

端子の名前を変更するためには、ESC キーを押して通常のカーソルに戻した後に、回路図上の INPUT/OUTPUT シンボルをダブル・クリックします。



Pin name 欄にわかりやすい名前を入力します。

エンティティの回路図入力作業は終わったら、「File」メニューから「Save」を実行して回路を保存してください。



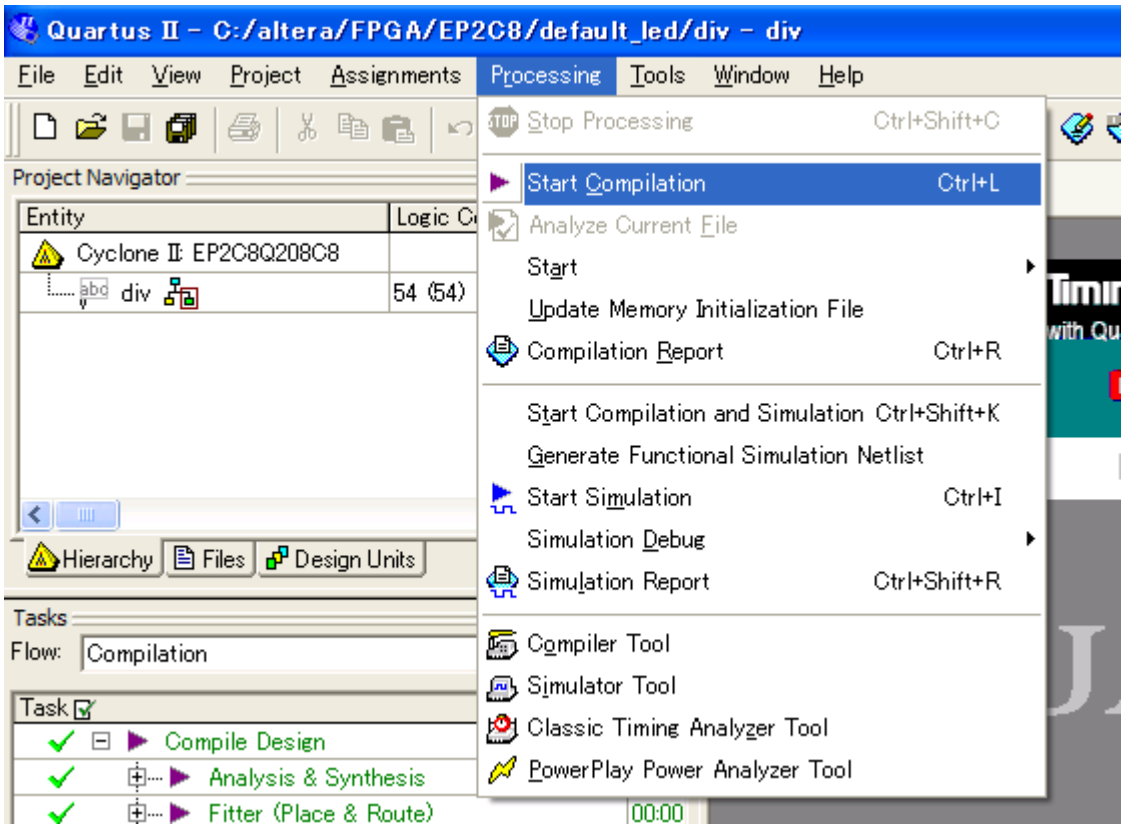
ある回路図の様子。

4.2 書き込み前の二つの作業

4.2.1 回路図をコンパイルする

入力した回路図から CPLD/FPGA に書き込むデータを生成するためには、作成した回路図をコンパイルしなければなりません。

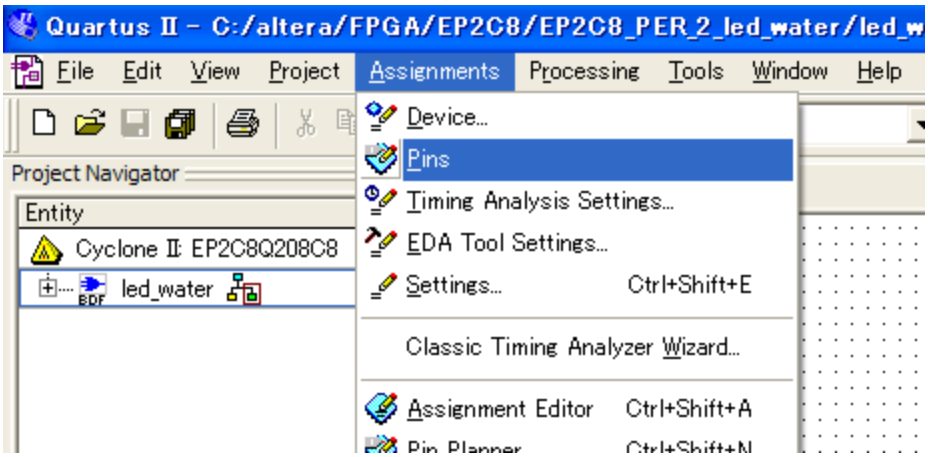
Quartus II の「Processing」メニューから「Start Compilation」を選択します。するとコンパイル処理が始まり、プロブレス・バーが働き始めます。コンパイルは数十秒で終了します。



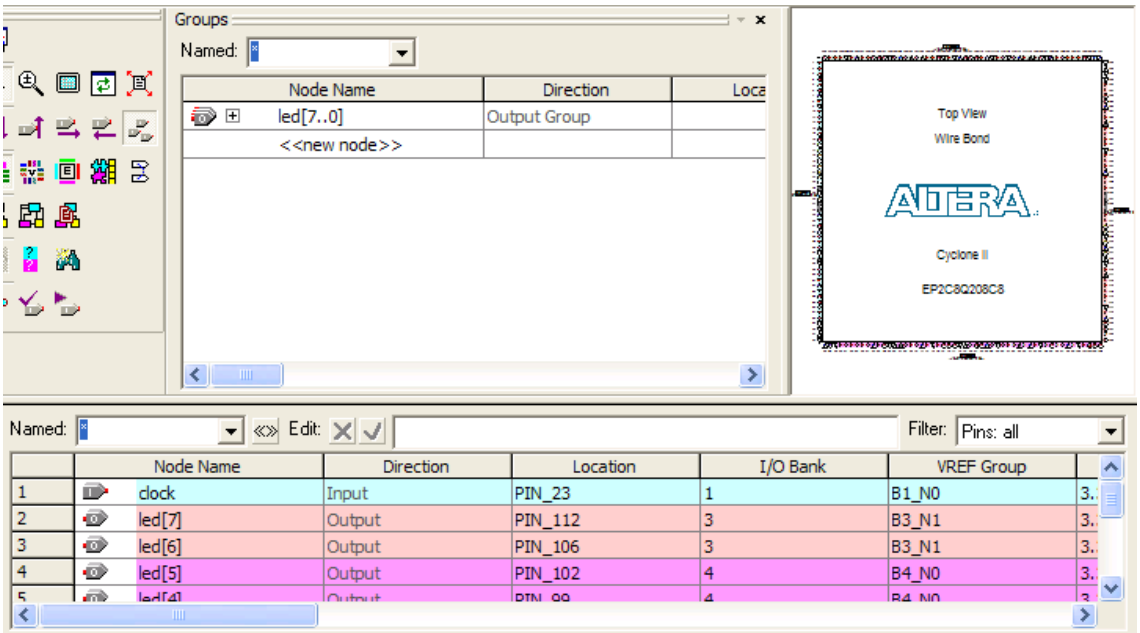
正常にコンパイルが終了しなっかた場合は、何か手順を間違えているか、回路図の入力をミスしている可能性があります。画面に表示されたメッセージを読めば、どのようなミスがあるかある程度知ることができます。

4.2.2 回路図の入出力と CPLD/FPGA の端子を関連づける

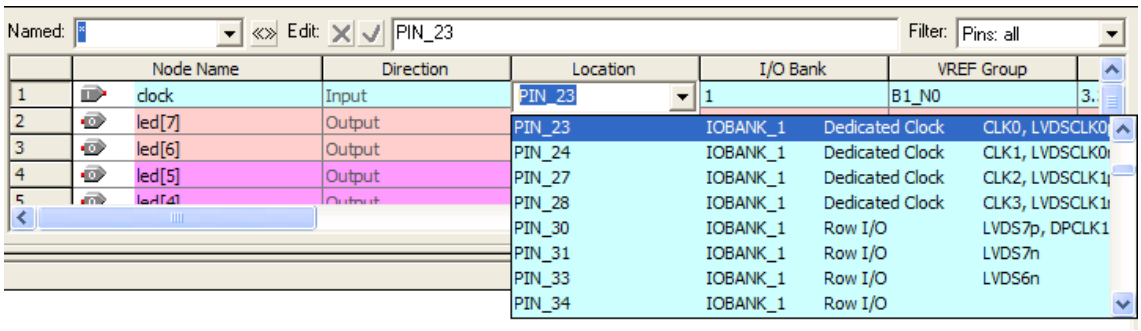
回路図上の端子名と CPLD/FPGA のピン番号との対応をピン・アサインと言います。ピン・アサインは、「Assignments」メニューの「Pins」を選択します。



Assignments Editor というウィンドウが開きます。



指定したいのはピンの場所、つまり Location です。行の Location の列をダブル・クリックして、ピンの番号を選択します。



ピン・アサインが完成したら、「File」メニューから「Save Project」を選択してプロジェクト全体を保存して、再度コンパイルを実行します。

コンパイル成功すれば、書き込むデータ*pof 又は*.sof を生成します。データを CPLD/FPGA に書き込みましょう。

※ 提供されたサンプルの一部又は全部、回路図で作成したものではありません。VHDL 又は Verilog という HDL(Hardware Description Language: ハードウェア記述言語)を使って、作成しました。VHDL/Verilog の使い方はほかの資料を参照してください。

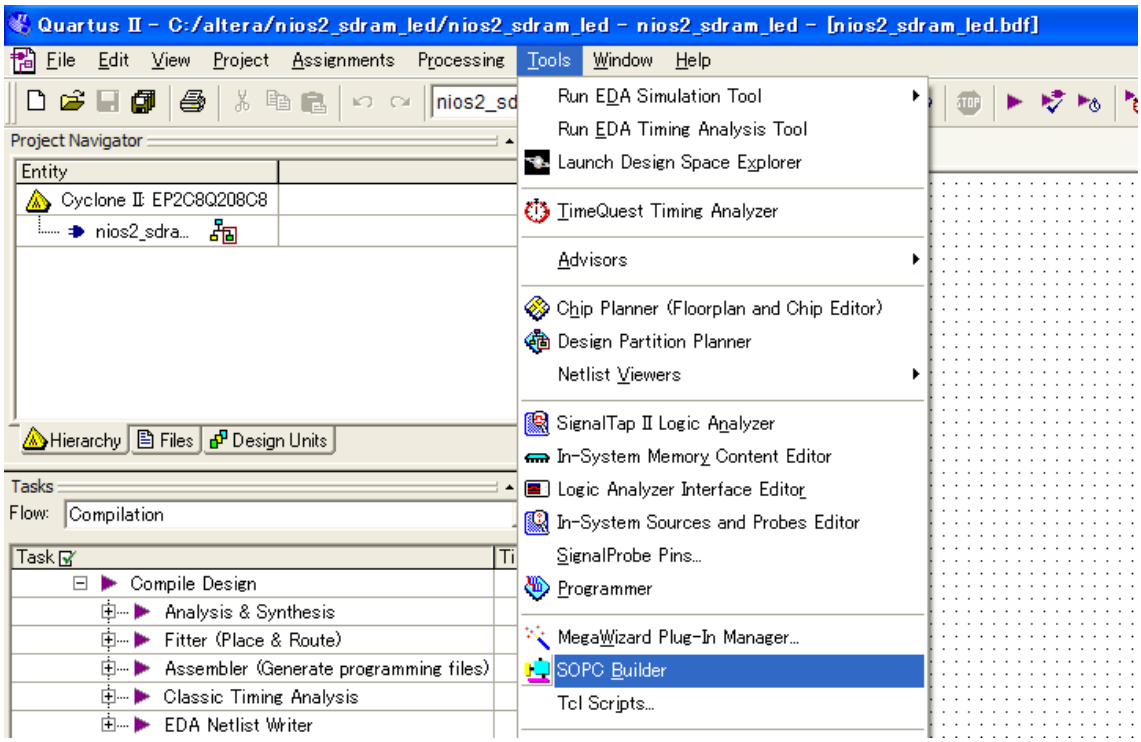
第五章 NIOS II システム・モジュールの設計

Cyclone II シリーズ FPGA はソフトプロセッサ NIOS II システムを搭載できます。NIOS II は、32bit CPU、命令・データキャッシュ搭載、最大 250MHz 動作します。とてもハイパフォーマンスなＣＰＵです。開発環境は、ポピューラーなＧＣＣで、無償提供されています。開発環境が信じられないほど簡単です。Ｈ８・ＰＩＣ を、お使いの方に使い比べて頂きたいシステムです。

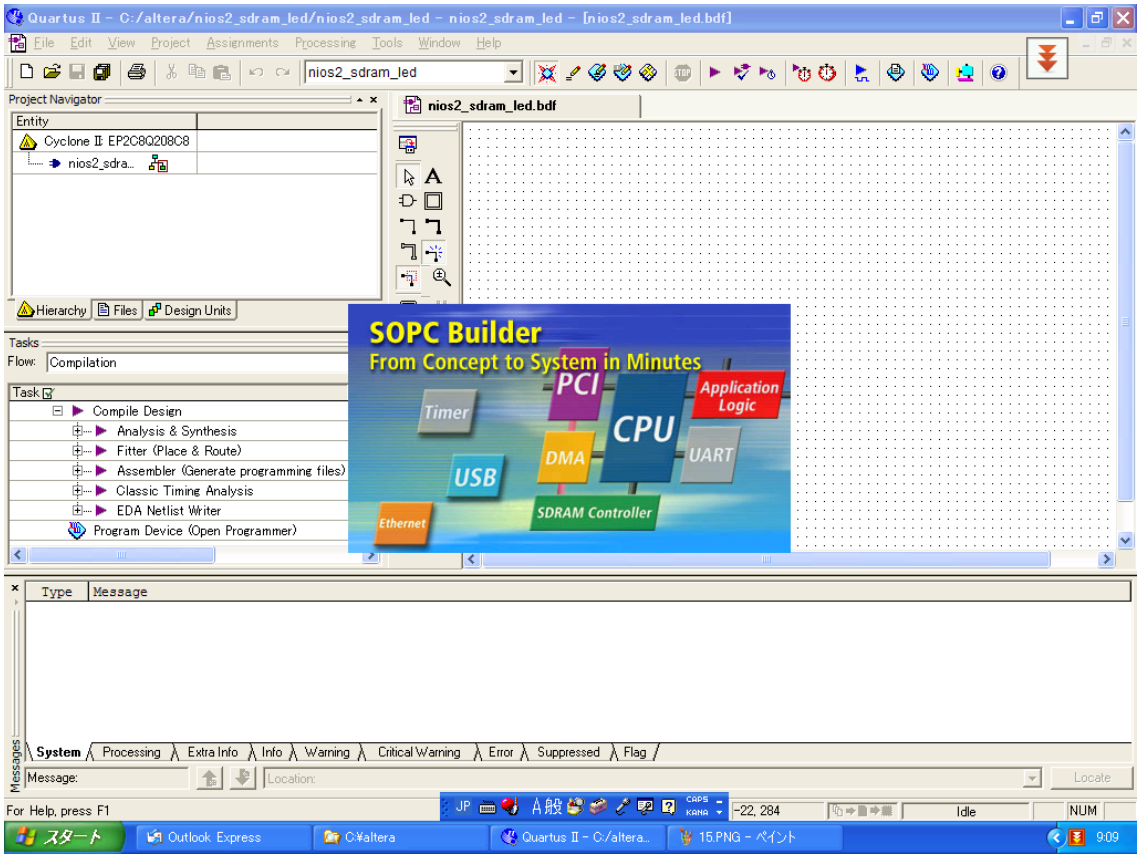
第四章の CPLD/FPGA の開発入門は定番シンボル(例えば 74 シリーズロジックなど)で回路図を設計します。今回は回路図に CPU を載せます。

まず、Quartus II を起動して、第四章に基づいて、ある空のトップ・エンティティを作ります。

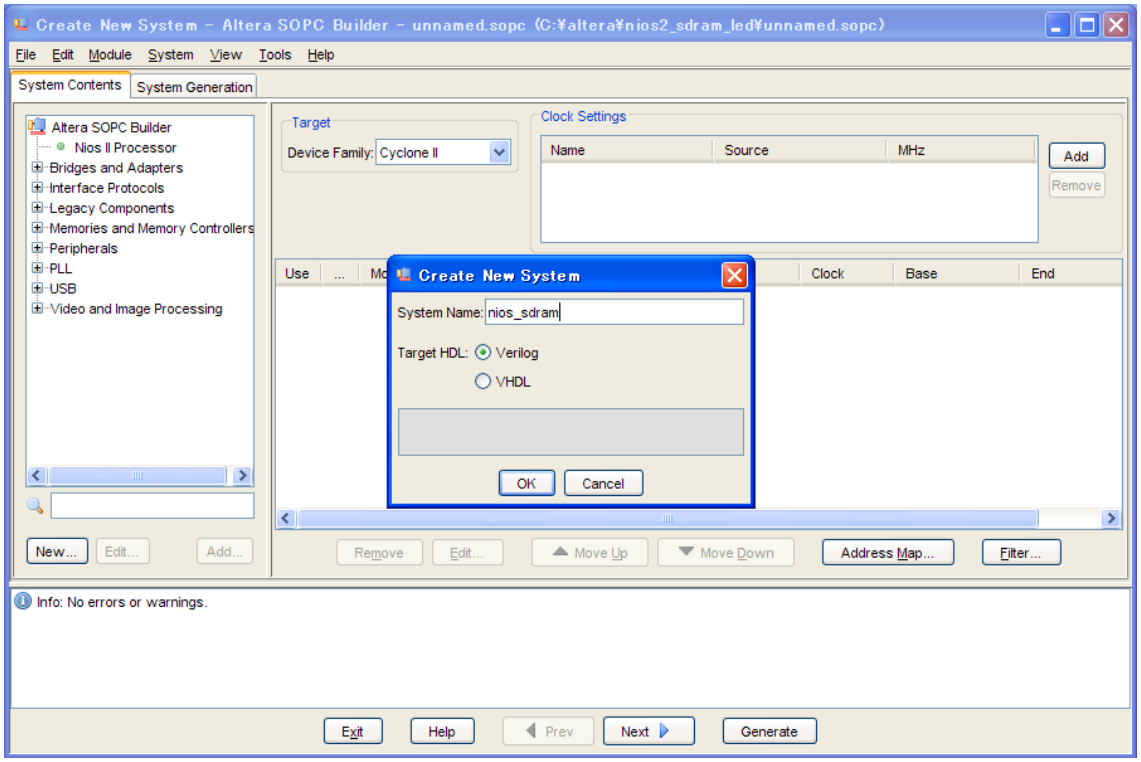
「Tools」メニューから「SOPC Builder」を選択し、SOPC Builder を起動します。


Quartus II - C:/altera/nios2_sdram_led/nios2_sdram_led - nios2_sdram_led - [nios2_sdram_led.bdf]
File Edit View Project Assignments Processing Tools Window Help
Run EDA Simulation Tool
Run EDA Timing Analysis Tool
Launch Design Space Explorer
TimeQuest Timing Analyzer
Advisors
Chip Planner (Floorplan and Chip Editor)
Design Partition Planner
Netlist Viewers
SignalTap II Logic Analyzer
In-System Memory Content Editor
Logic Analyzer Interface Editor
In-System Sources and Probes Editor
SignalProbe Pins...
Programmer
MegaWizard Plug-In Manager...
SOPC Builder
Tcl Scripts...
Project Navigator
Entity
Cyclone II: EP2C8Q208C8
Hierarchy Files Design Units
Tasks
Flow: Compilation
Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programming files)
Classic Timing Analysis
EDA Netlist Writer

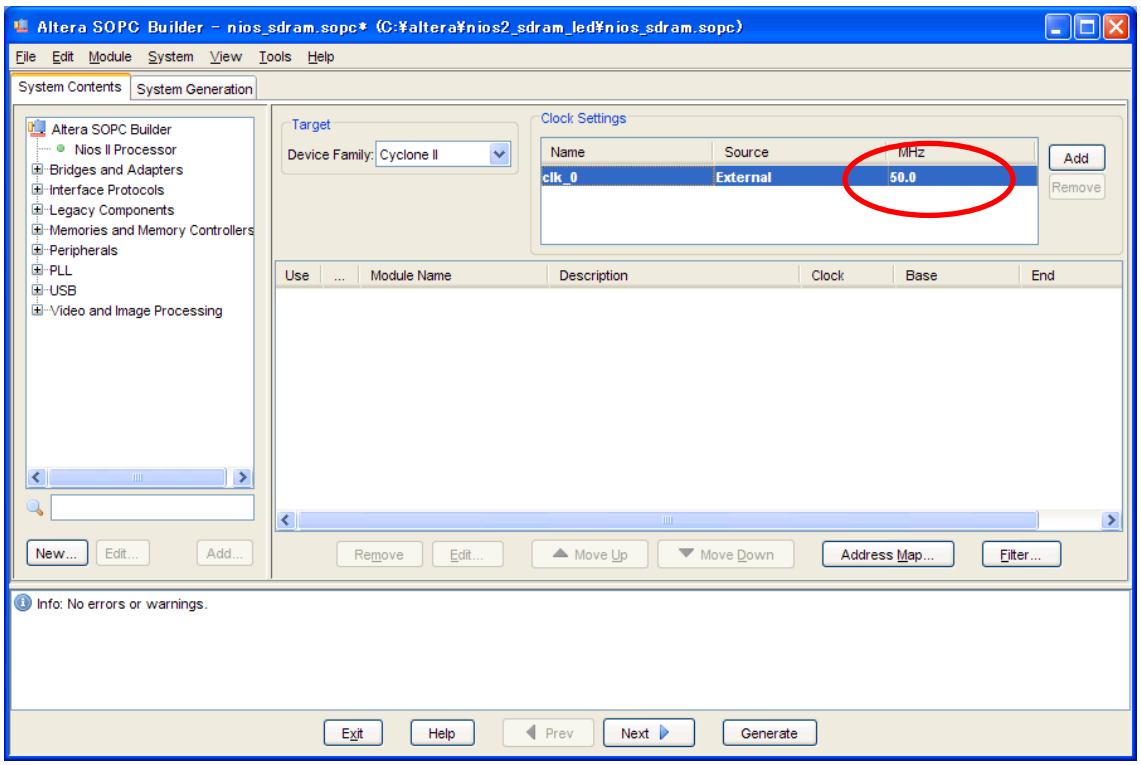
起動の様子です。


SOPC Builder
From Concept to System in Minutes
PCI
Application Logic
Timer
CPU
DMA
UART
USB
Ethernet
SDRAM Controller
Program Device (Open Programmer)
Type Message
System Processing Extra Info Info Warning Critical Warning Error Suppressed Flag
For Help, press F1

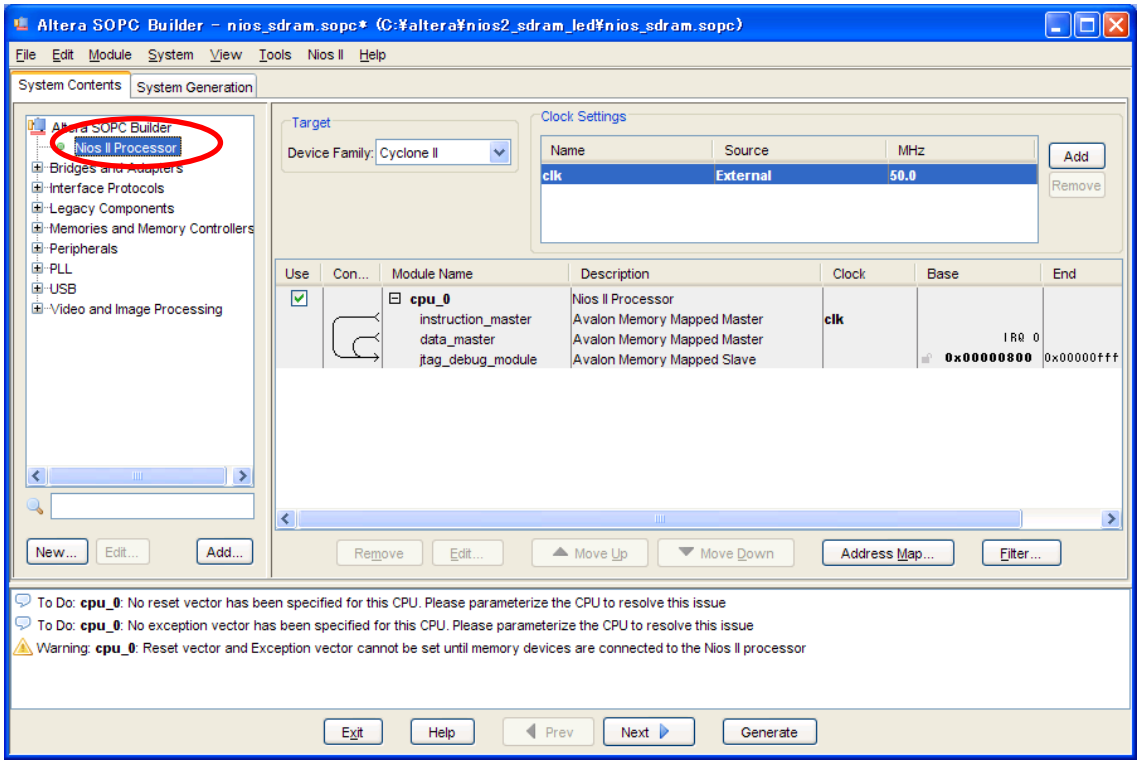
ダイアログに NIOS II システムの名前「nios_sram」を入力してください。名前は必ずトップ・エンティティの名前と異なります。



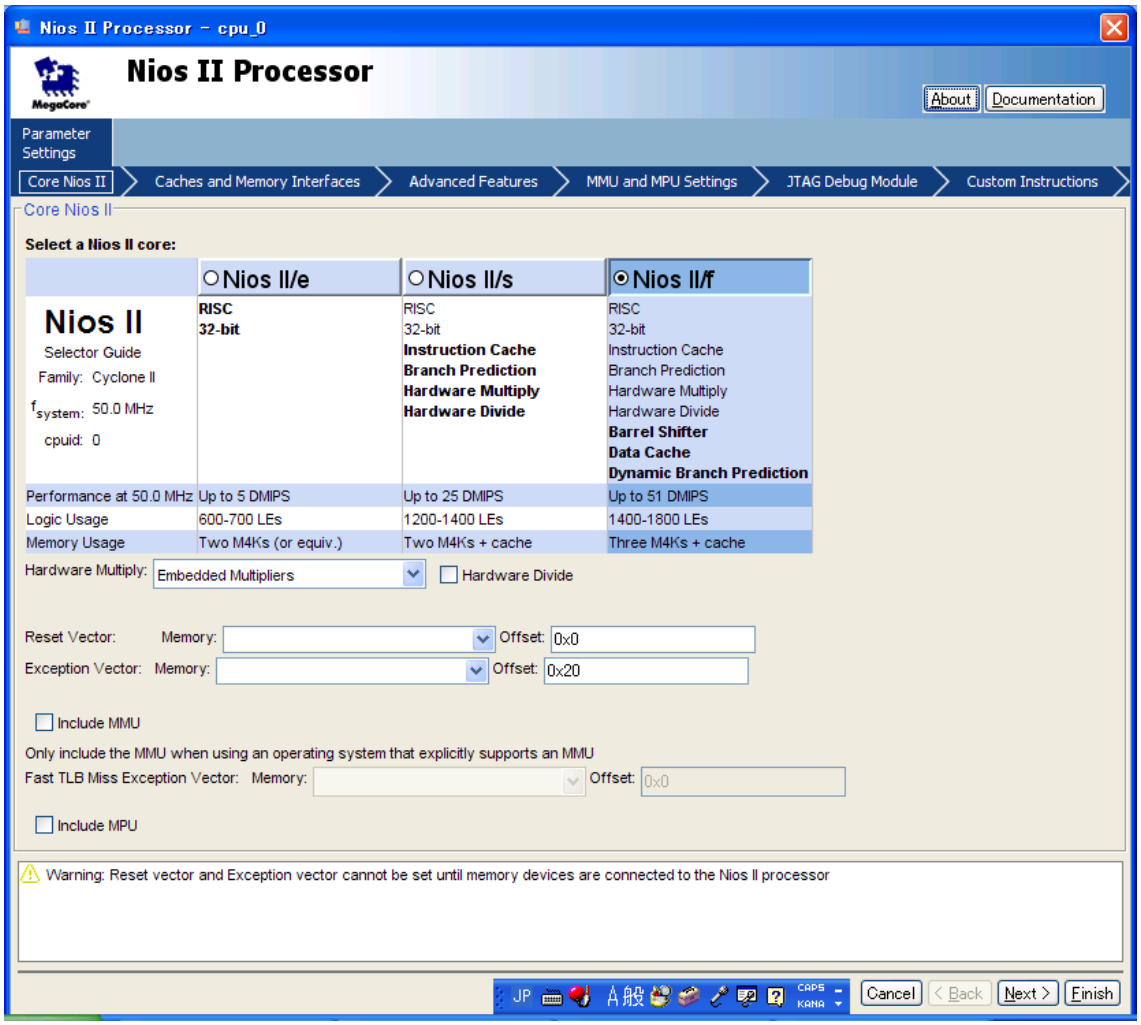
システムの周波数を入力します。今回の例は 50MHz です。周波数の欄でダブル・クリックして、周波数を入力できます。



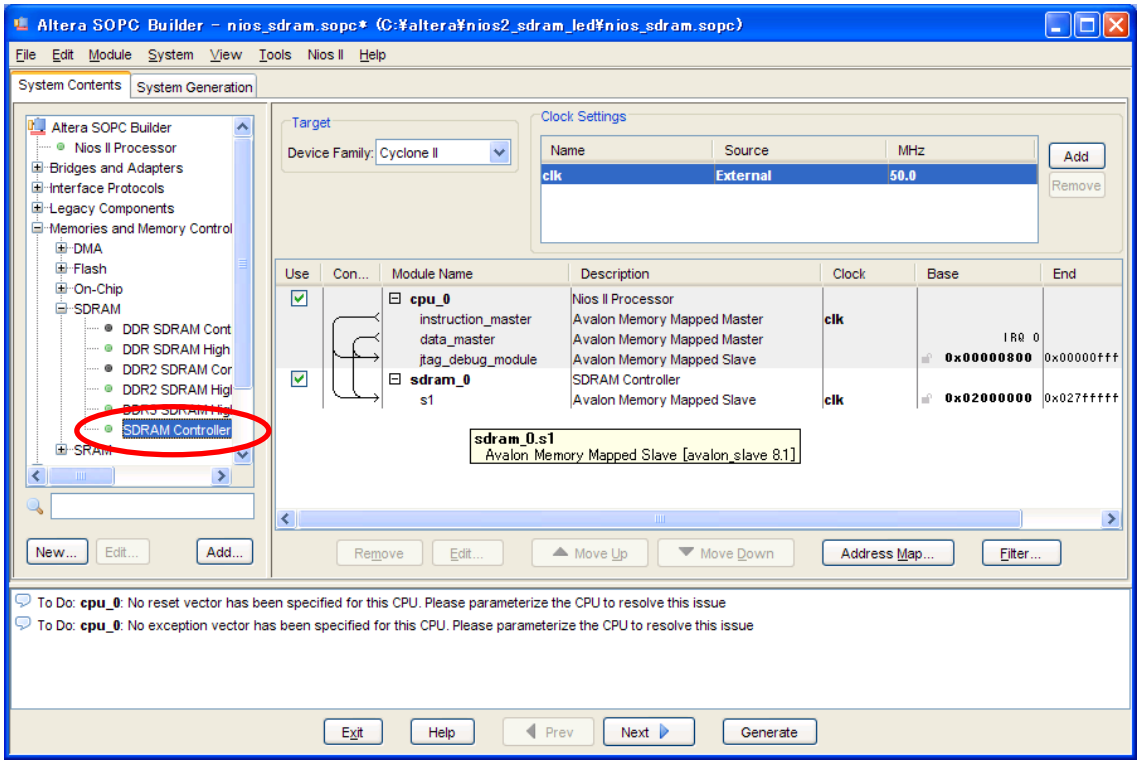
左側の「Nios II Processor」をダブル・クリックして、Nios II CPU コアを添加します。



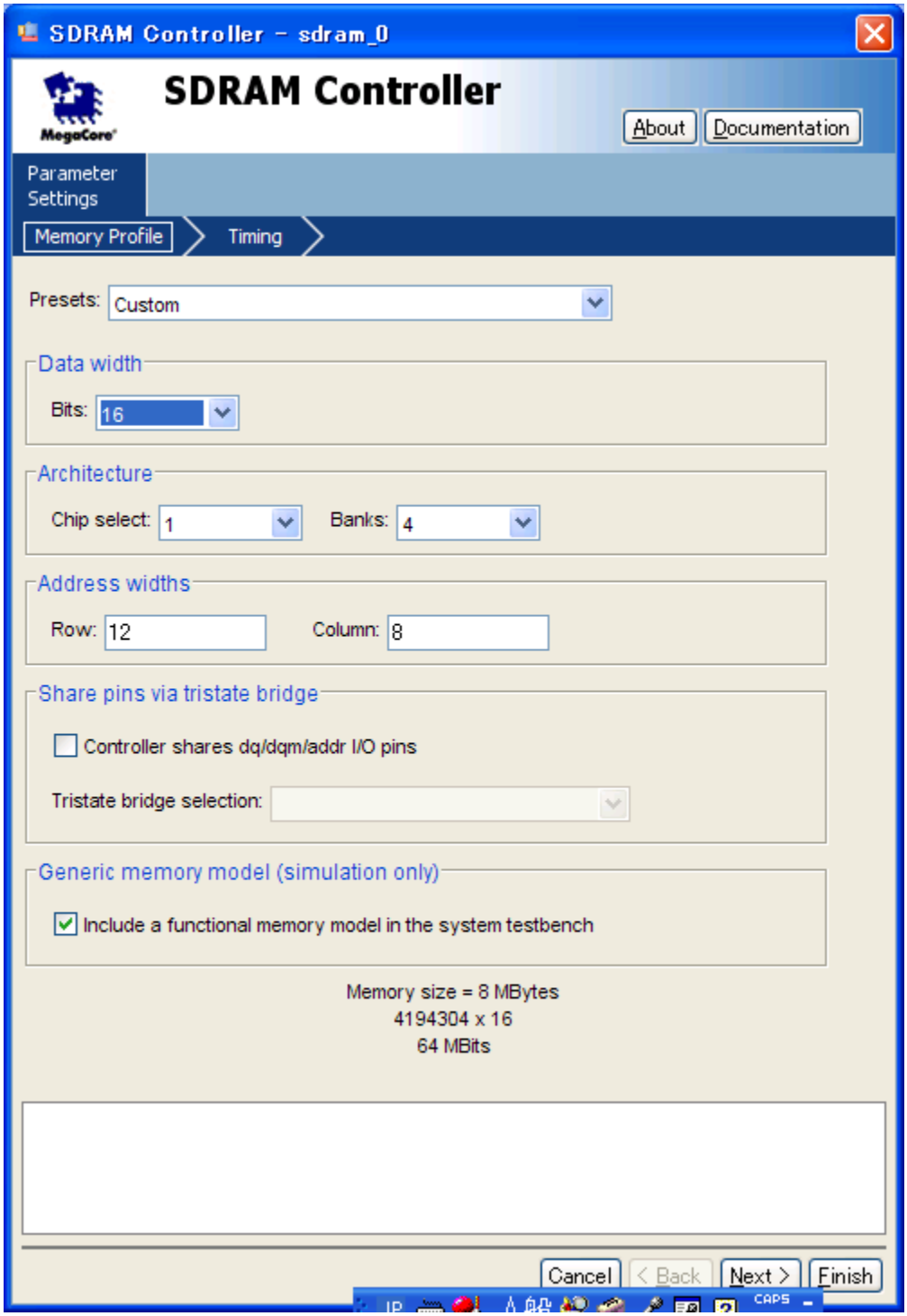
Nios II は性能の違う 3 種類の CPU コアが提供されていますが、今回はデフォルトの高性能のグレート(Nios II/f)を使用します。このグレードは、デフォルトでハードウェア乗算器と命令・データキャッシュ、JTAG デバッガが組み込まれています。今回はこれをそのまま使用します。「Finish」ボタンを押してください。



EP2C8 基板の SDRAM を使うため、SDRAM コントローラを Nios II システムに組み込みます。左側の「SDRAM Controller」を選択し、ダブル・クリックして SDRAM コントローラを添加します。

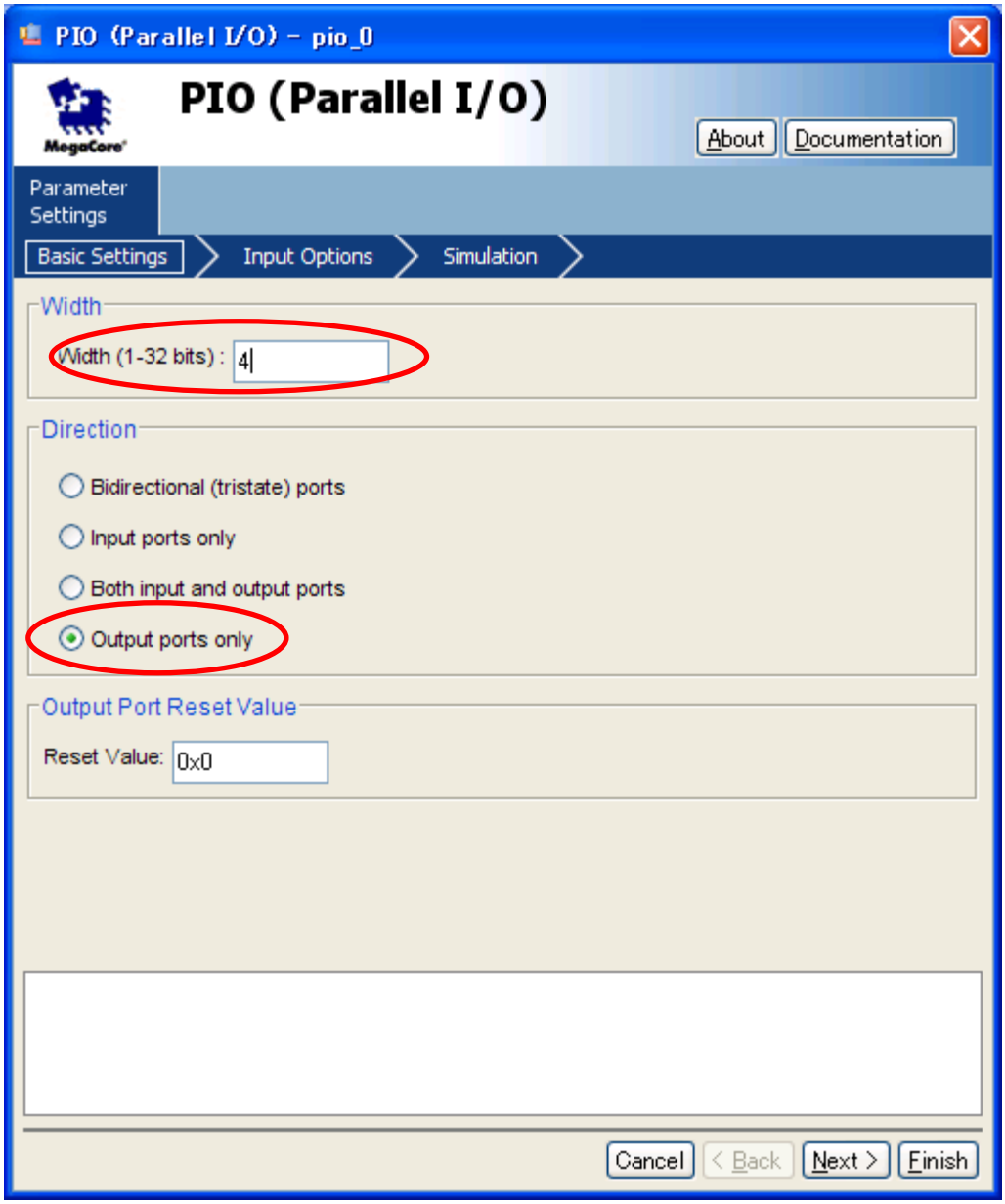


それぞれのメモリごとにデータのビット幅、チップ・セレクトの本数、バンク数、アドレスのロウ・カラムの本数、および AC スパックをデータ・シートから読み取り、間違いのないように入力します。



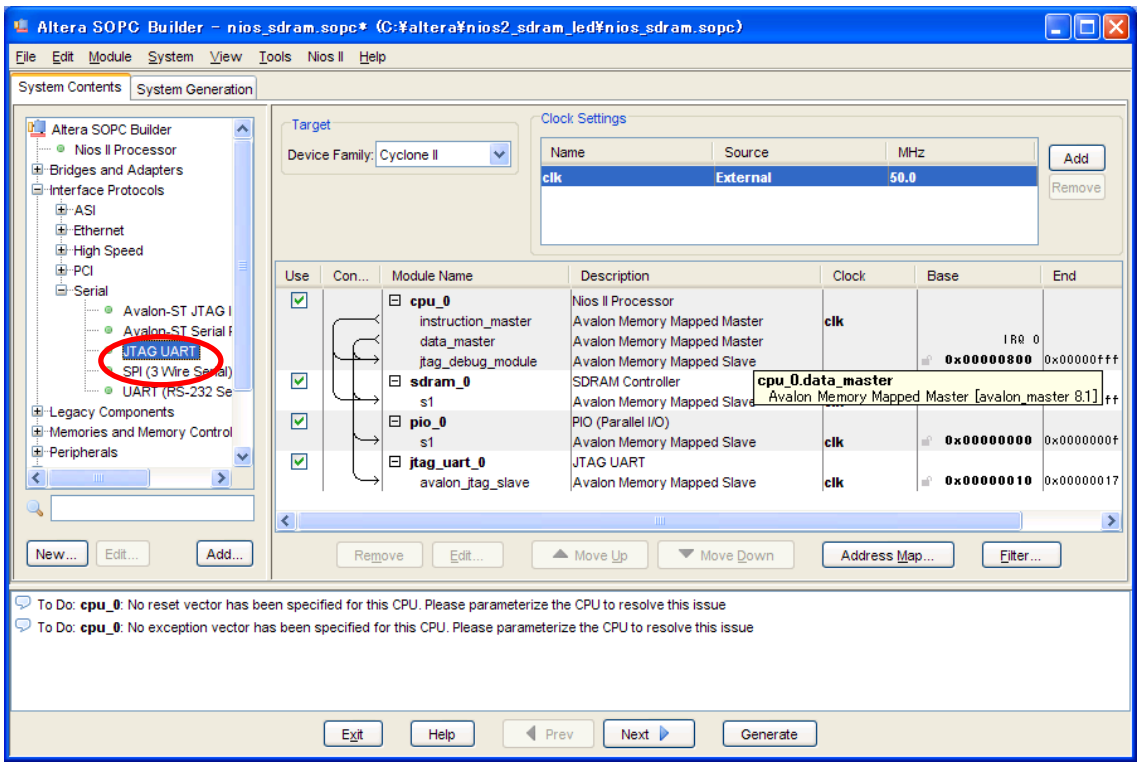
EP2C8 基板は 4 ビットの LED があります。LED を制御するポートを Nios II システムに

組み込みます。左側の「PIO(Parallel I/O)」を選択し、ダブル・クリックして LED コントローラを添加します。



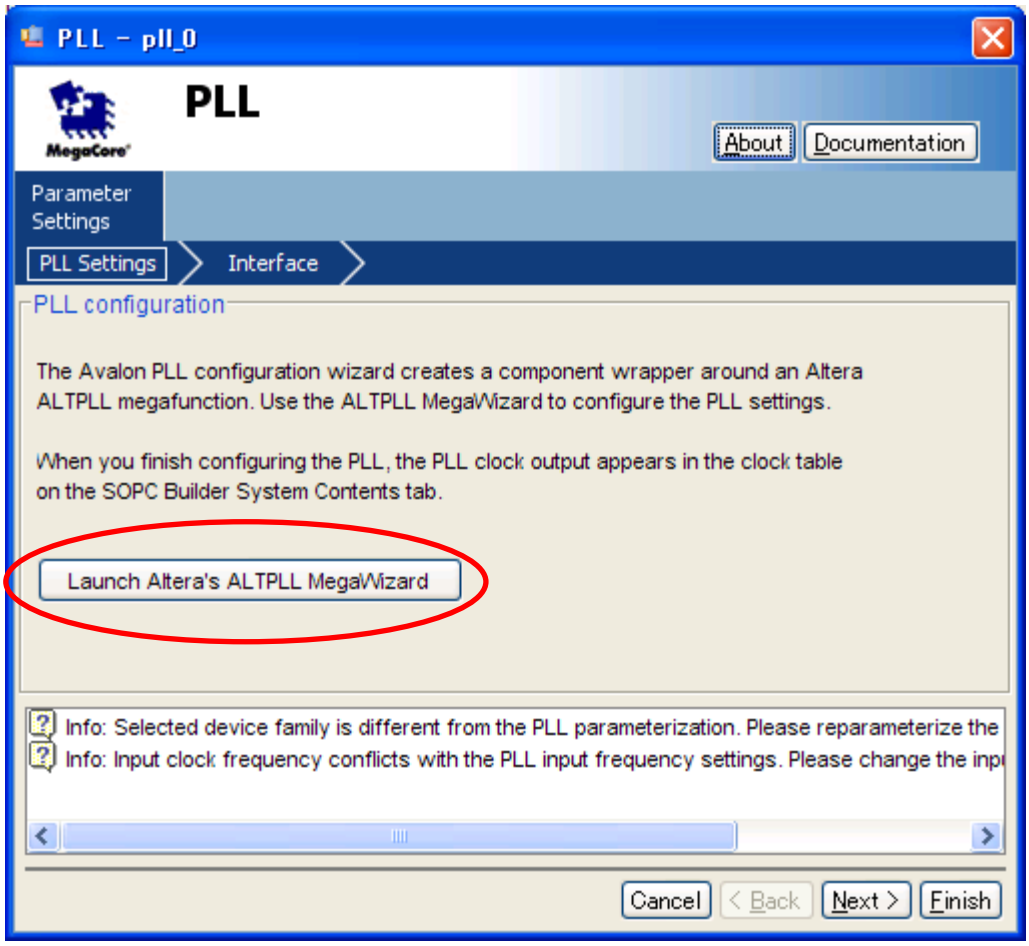
ビット幅に 4 を入力してください。「Output ports only」を選択します。

ホストと会話するため、JTAG-UART も組み込むことが必要です。左側の「JTAG_UART」を選択し、ダブル・クリックして JTAG-UART を添加します。

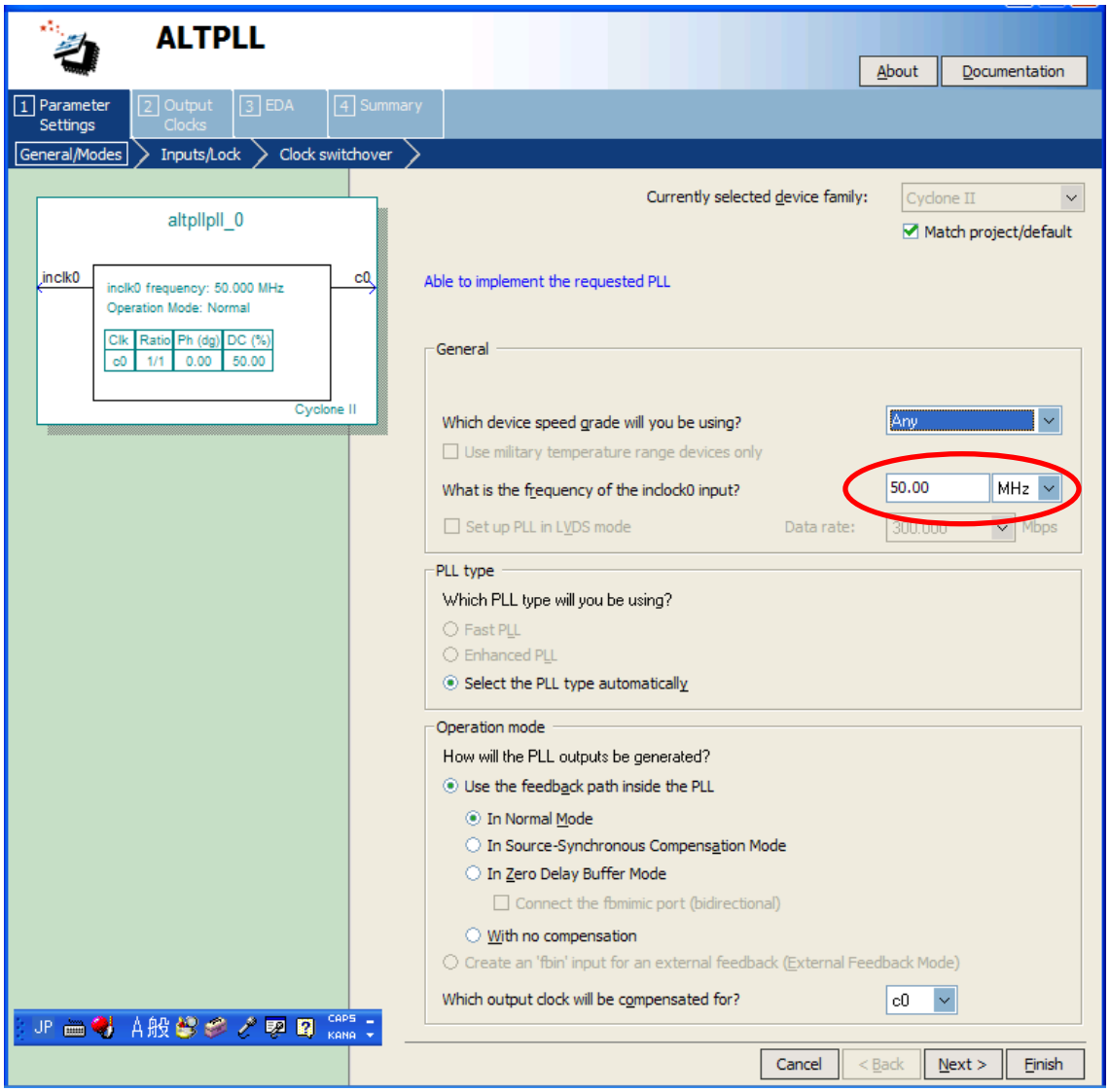

Altera SOPC Builder - nios_sdram.sopc* (C:¥altera¥nios2_sdram_led¥nios_sdram.sopc)
File Edit Module System View Tools Nios II Help
System Contents
System Generation
Altera SOPC Builder
Nios II Processor
Bridges and Adapters
Interface Protocols
ASI
Ethernet
High Speed
PCI
Serial
Avalon-ST JTAG I
Avalon-ST Serial F
JTAG UART
SPI (3 Wire Serial)
UART (RS-232 Se
Legacy Components
Memories and Memory Control
Peripherals
New...
Edit...
Add...
Target
Device Family: Cyclone II
Clock Settings
Name
Source
MHz
clk
External
50.0
Add
Remove
Use
Con...
Module Name
Description
Clock
Base
End
cpu_0
Nios II Processor
instruction_master
Avalon Memory Mapped Master
clk
data_master
Avalon Memory Mapped Master
IRQ 0
jtag_debug_module
Avalon Memory Mapped Slave
0x00000800
0x00000fff
sdram_0
SDRAM Controller
cpu_0.data_master
Avalon Memory Mapped Master [avalon_master 8.1]
s1
Avalon Memory Mapped Slave
pio_0
PIO (Parallel I/O)
s1
Avalon Memory Mapped Slave
clk
0x00000000
0x0000000f
jtag_uart_0
JTAG UART
avalon_jtag_slave
Avalon Memory Mapped Slave
clk
0x00000010
0x00000017
Remove
Edit...
Move Up
Move Down
Address Map...
Filter...
To Do: cpu_0: No reset vector has been specified for this CPU. Please parameterize the CPU to resolve this issue
To Do: cpu_0: No exception vector has been specified for this CPU. Please parameterize the CPU to resolve this issue
Exit
Help
Prev
Next
Generate

デフォルトの設定のまま進んでください。

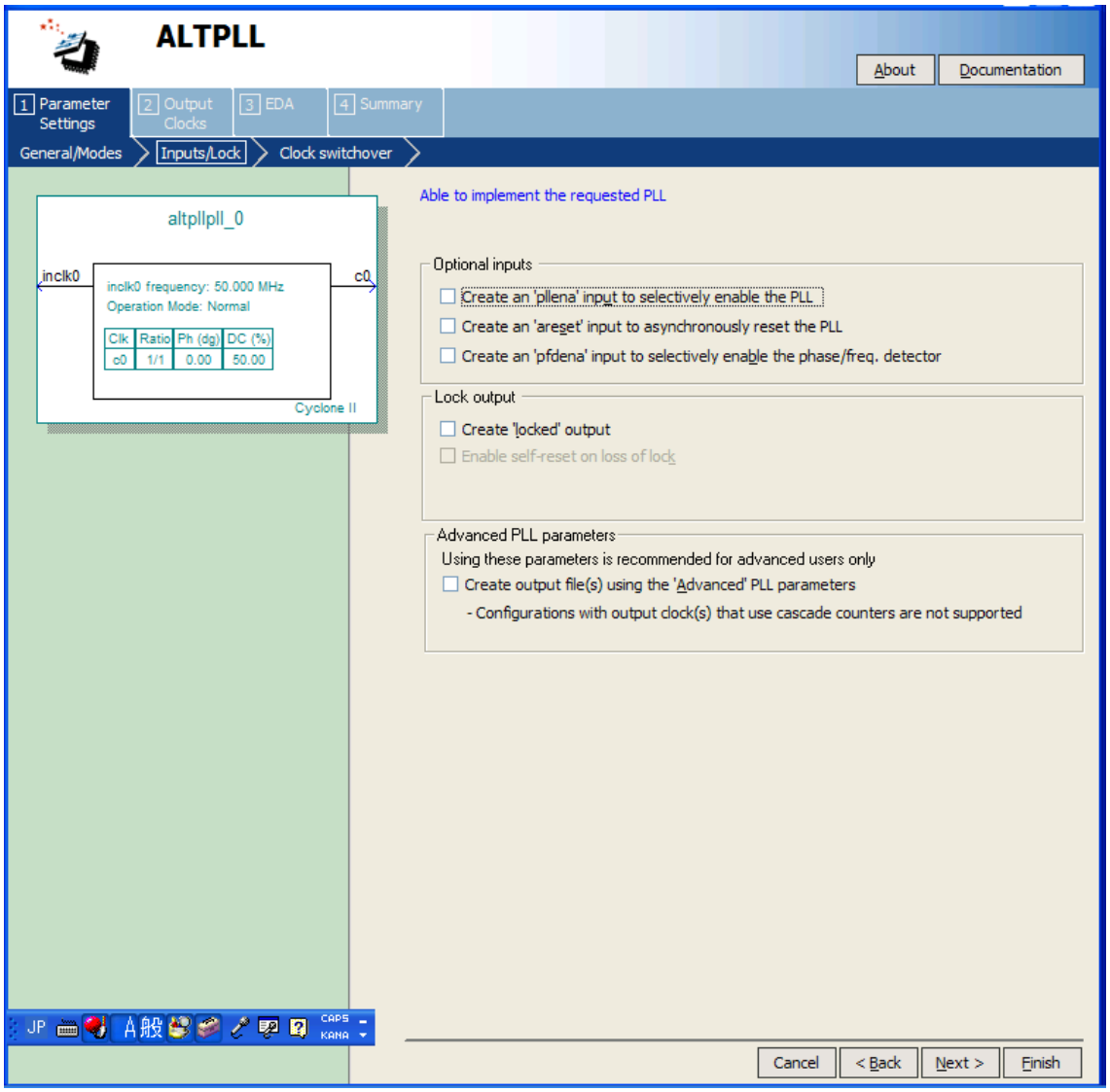
EP2C8 基板は一つの 50MHz の水晶発振器しかありません。ほかの周波数又は位相クロックのため、PLL を Nios II システムに組み込みます。左側の「PLL」を選択し、ダブル・クリックして PLL を添加します。



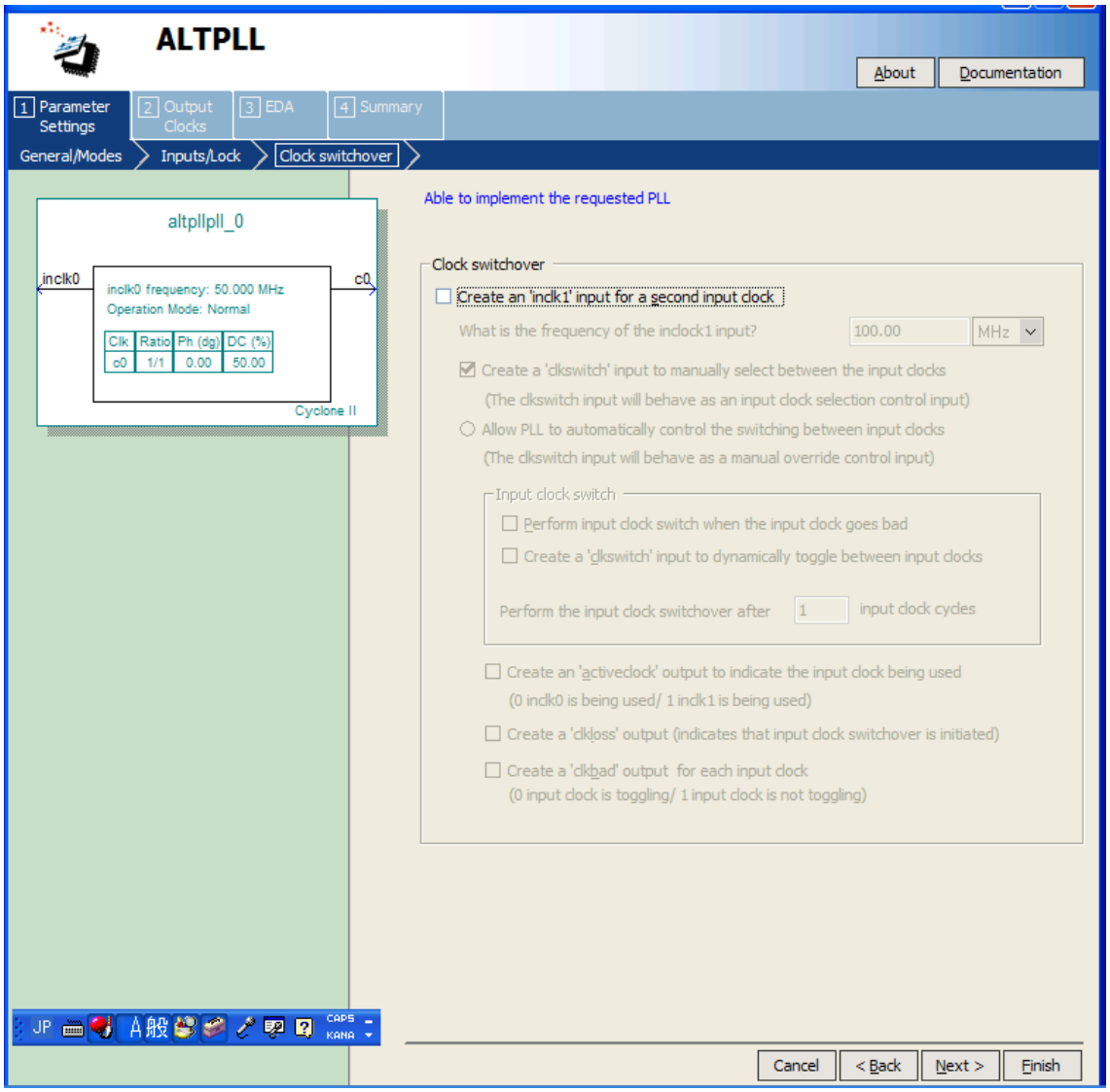
[Launch Altera's ALTPLL MegaWizard]ボタンを押して、PLL を設定します。



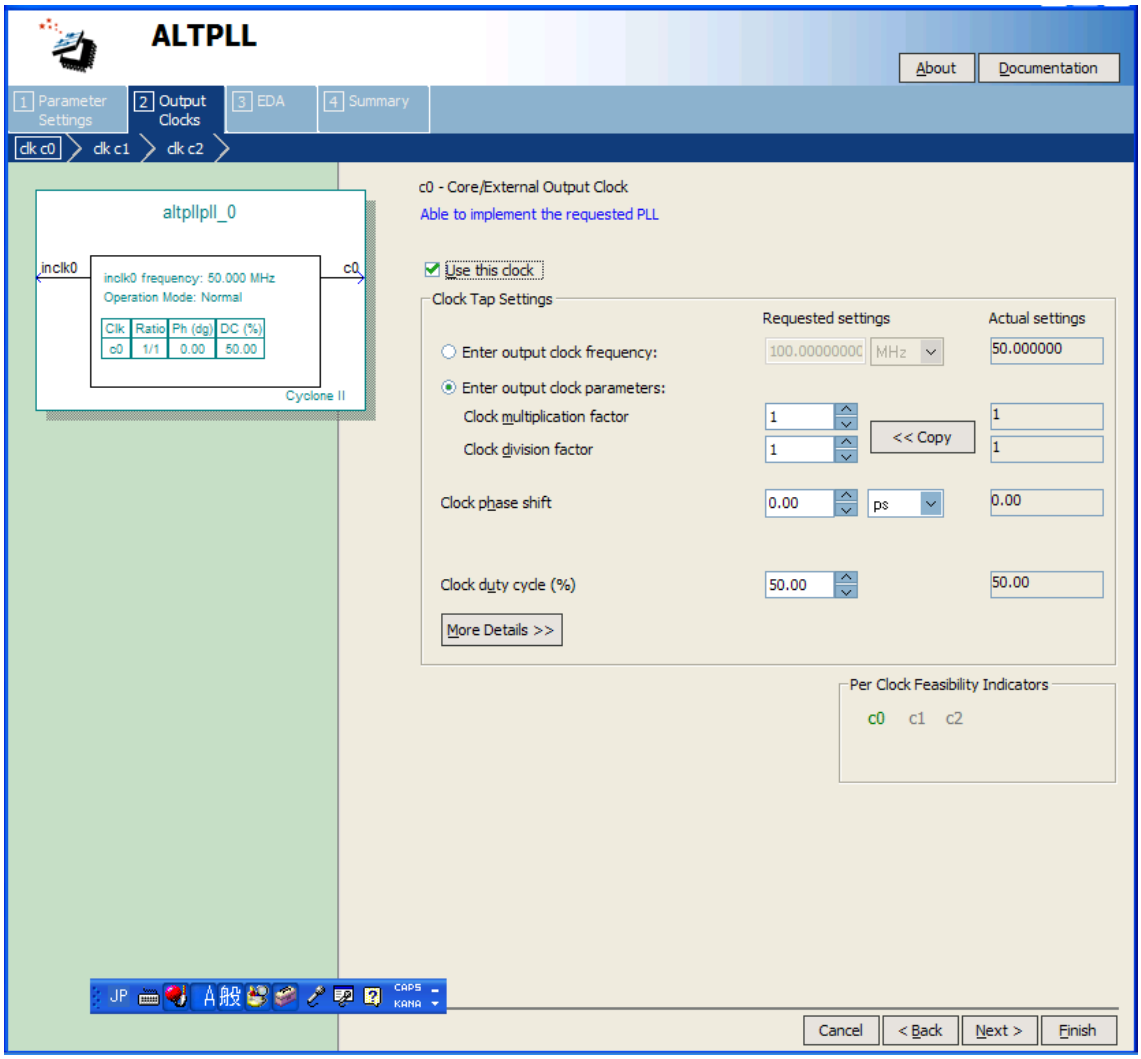
入力クロックは 50MHz を入力します。「Next」を押します。



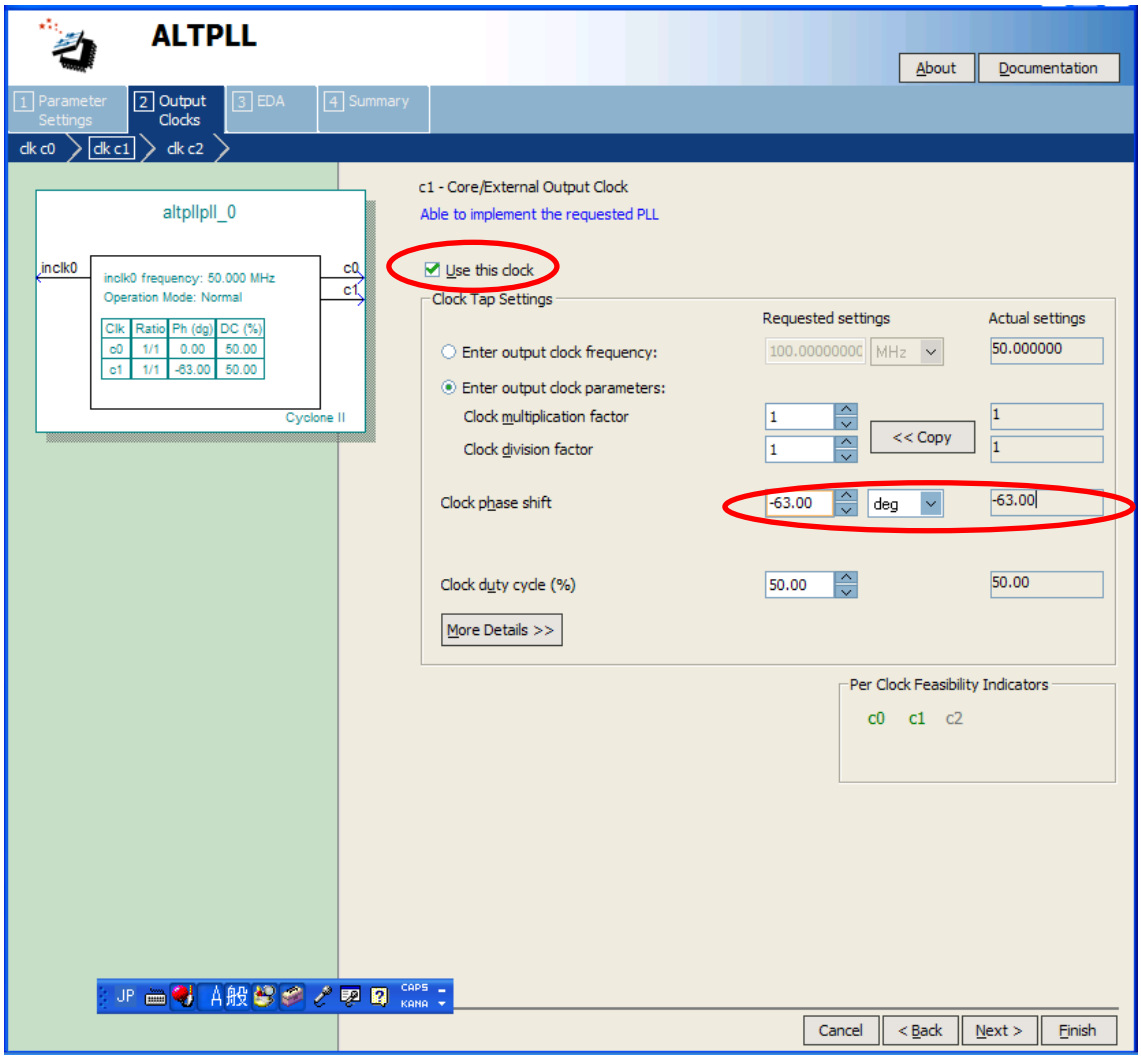
そのまま「Next」を押します。



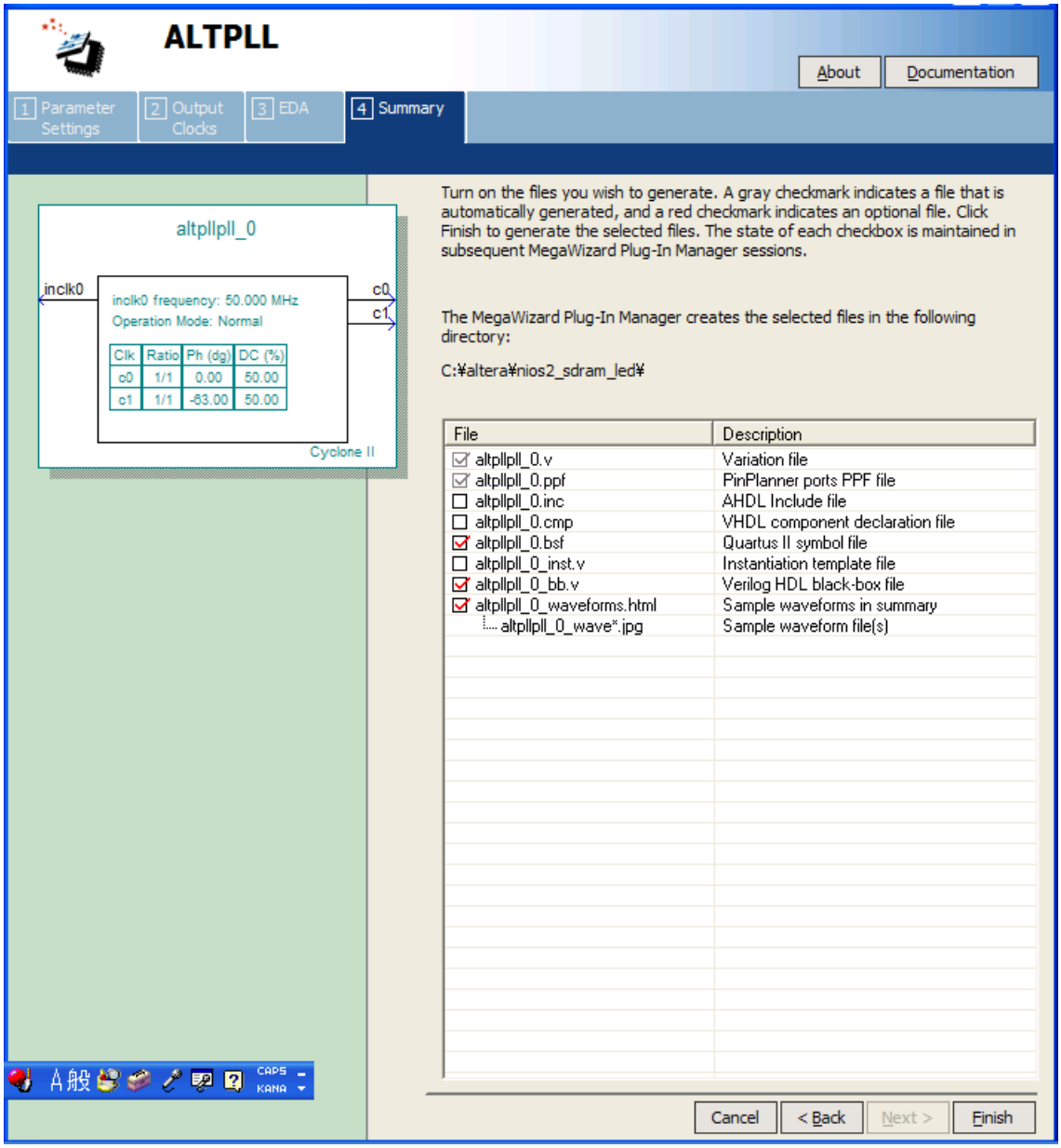
そのまま「Next」を押します。



そのまま「Next」を押します。

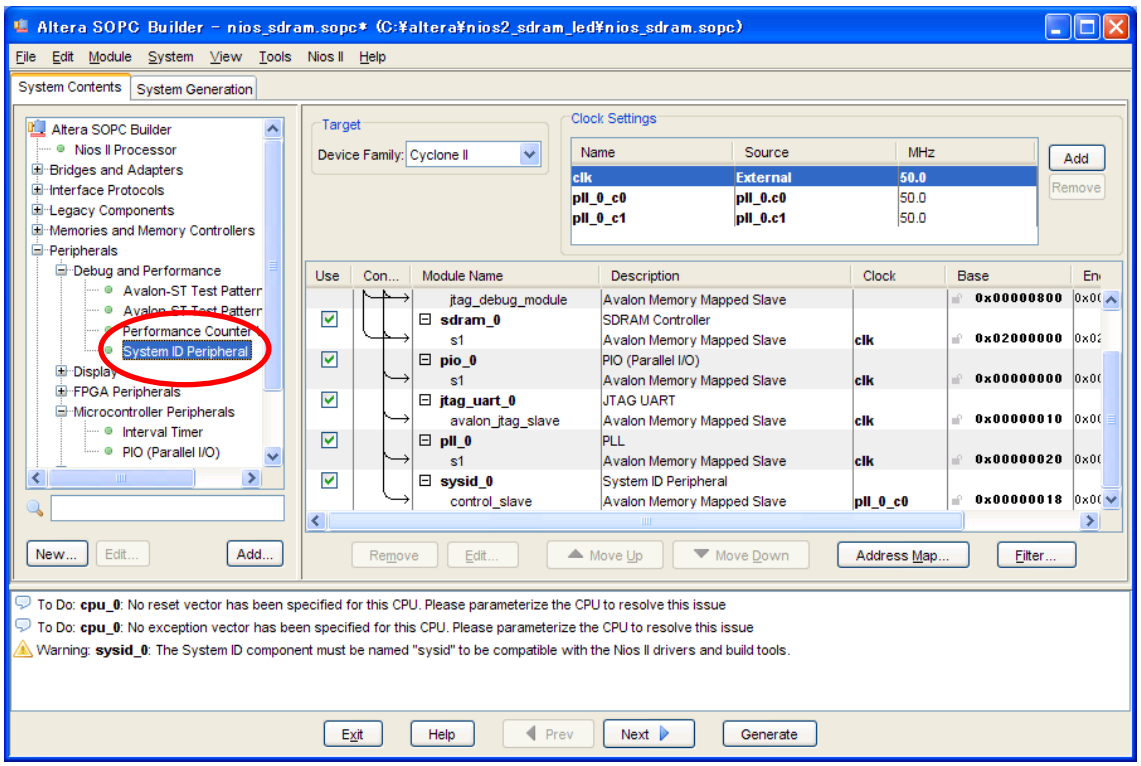


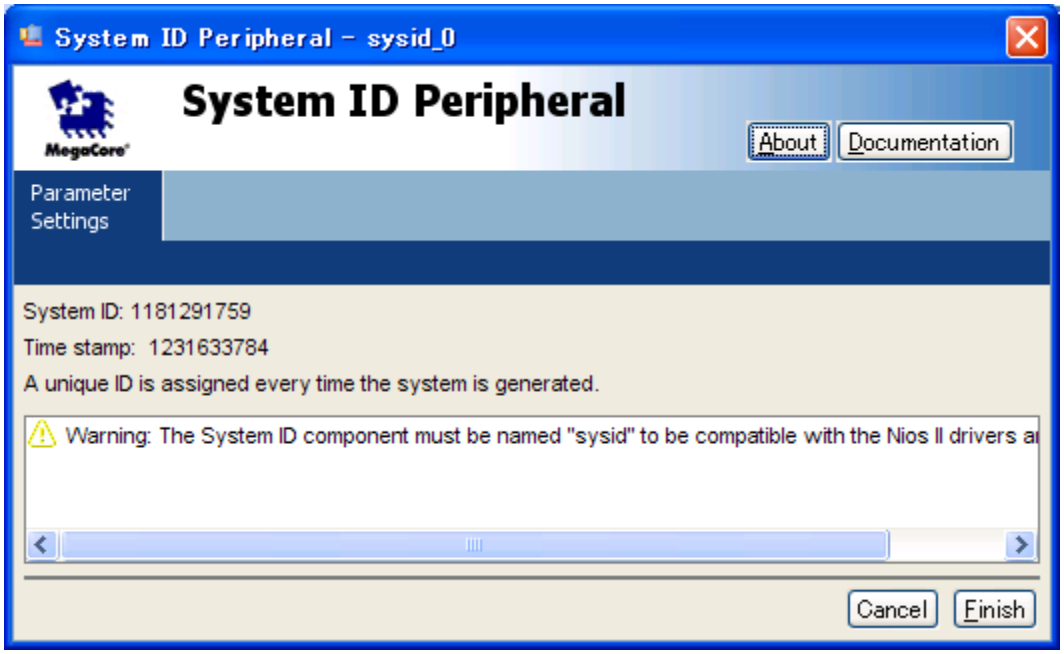
新しい出力クロックを添加します。クロックの位相は-63 に設定してください。このクロックは SDRAM 用の制御クロックです。「Next」を押します。



最後の確認です。「Finish」を押します。

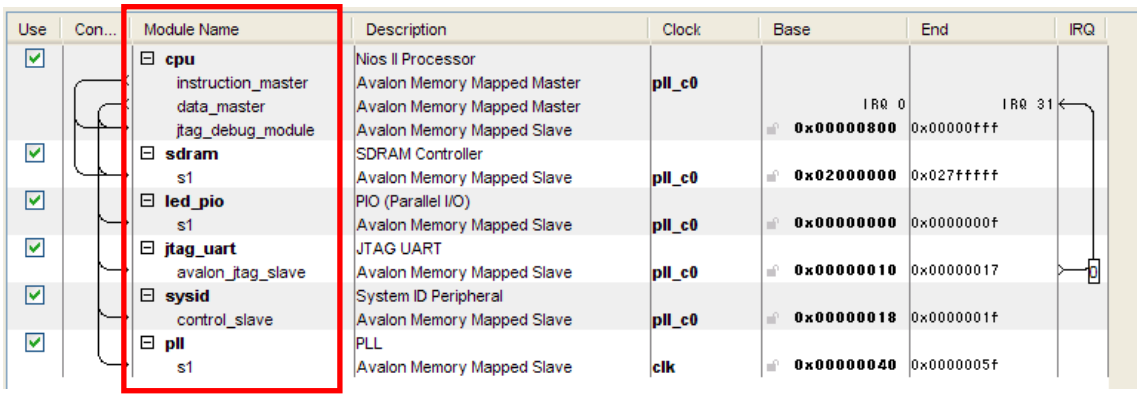
Sysid モジュールは、Altera 社のソフトウェア開発環境 IDE を用いてソフトウェアをダウンロードする際に、ハードウェアとの整合性の確認に利用する ID 情報を格納するモジュールです。ここで読み出せる値は、SOPC Builder でロジックを生成するときに与えられるものです。左側の「System ID Peripheral」を選択し、ダブル・クリックして Sysid を添加します。



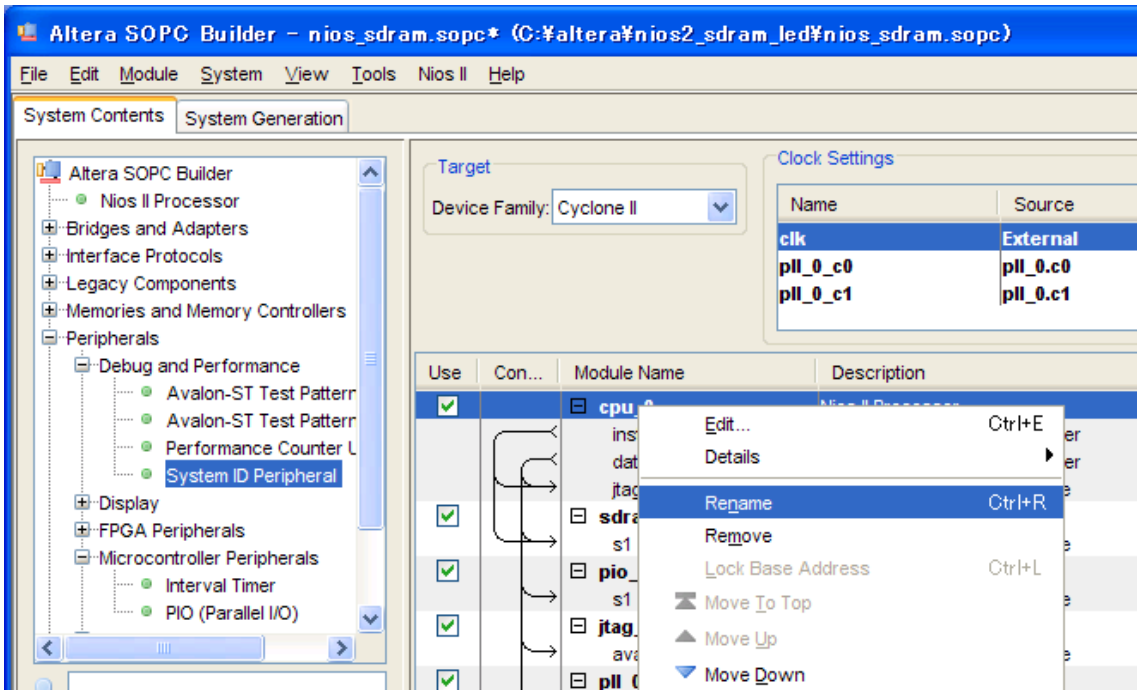


[Finish]を押して、Sysid を生成します。

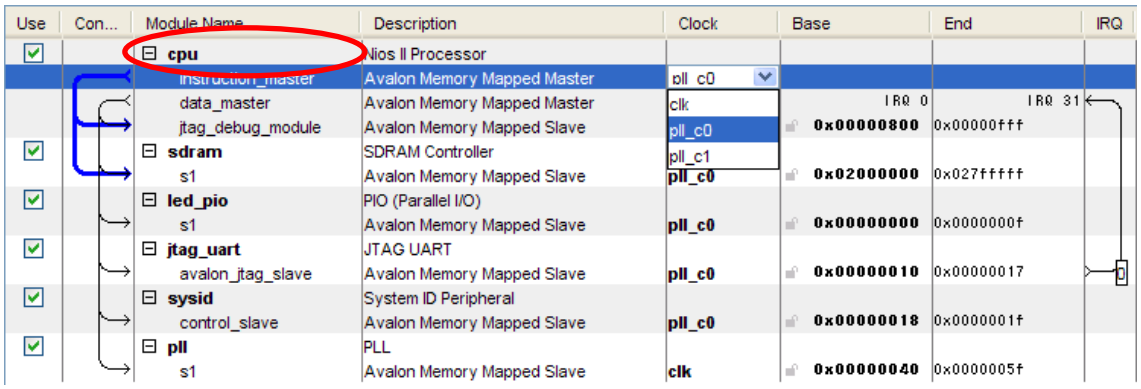
これまで、必要なモジュールをすべて組み込みました。



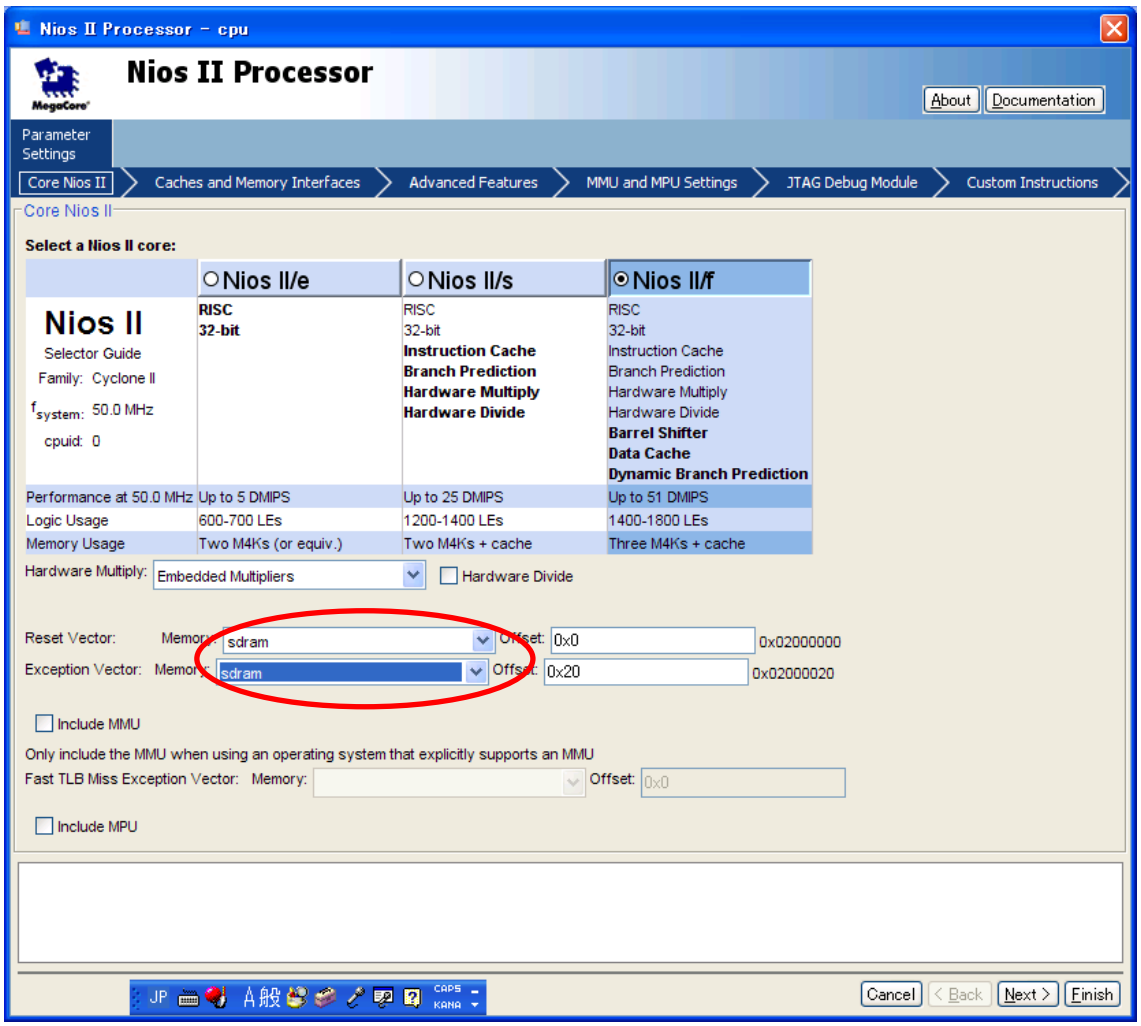
モジュールの名前はわかりやすい名前に変更します。モジュールの名前欄で右クリックして、出てきたメニューから「Rename」を選択します。



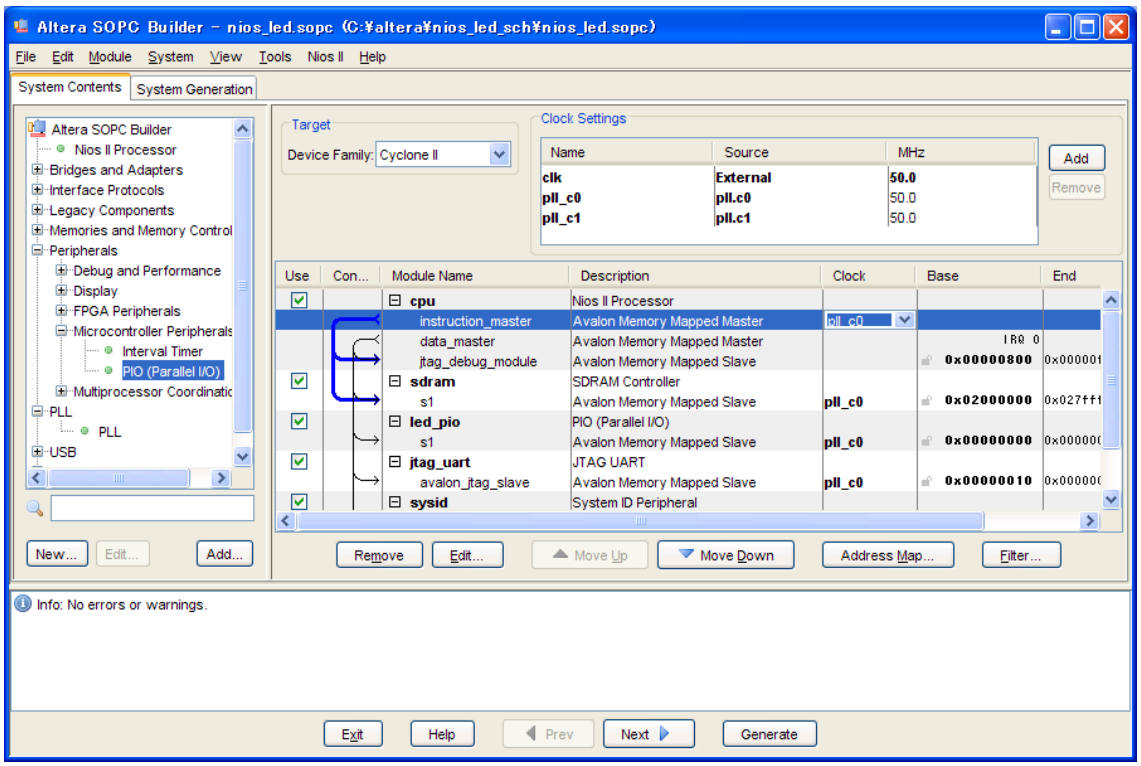
モジュール用のクロックを選択します。[Clock]欄でダブル・クリックして、PLL のクロック pll_c0 を選択します。PLL モジュールは外部クロック「clk」を使用します。



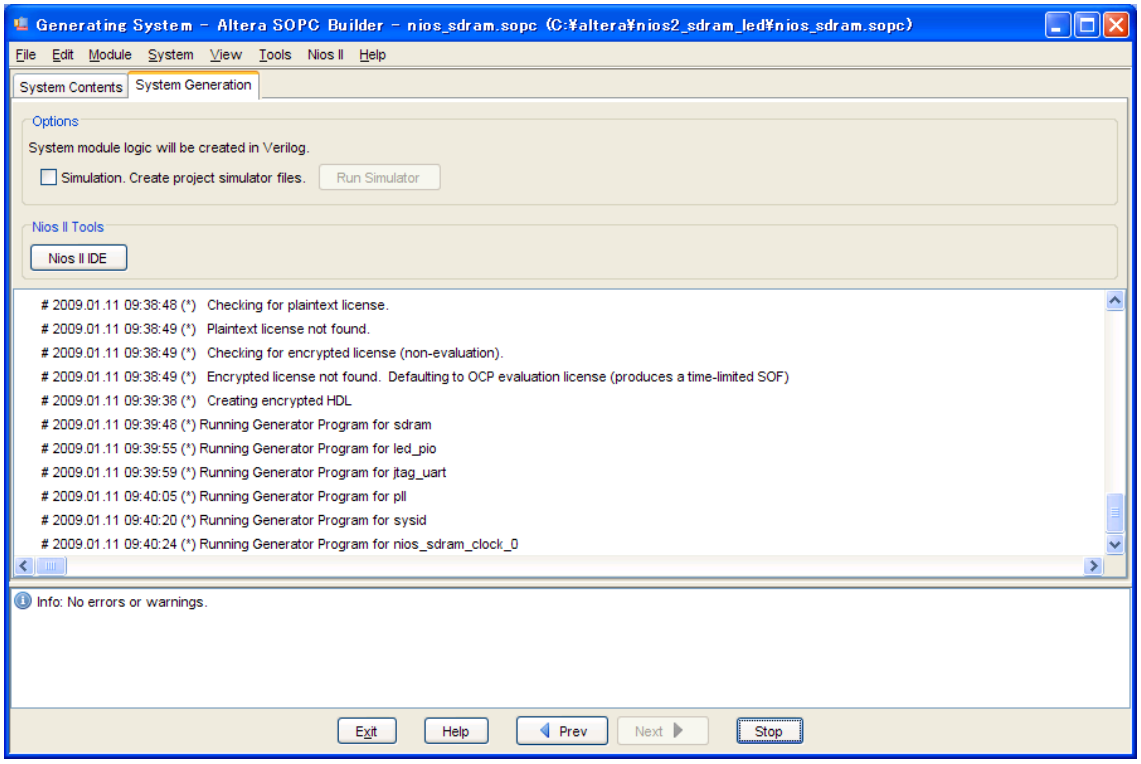
もう一つの設定は CPU のリセットと実行アドレスです。モジュール CPU の名前欄でダブル・クリックして、CPU 設定のダイアログを再び開きます。



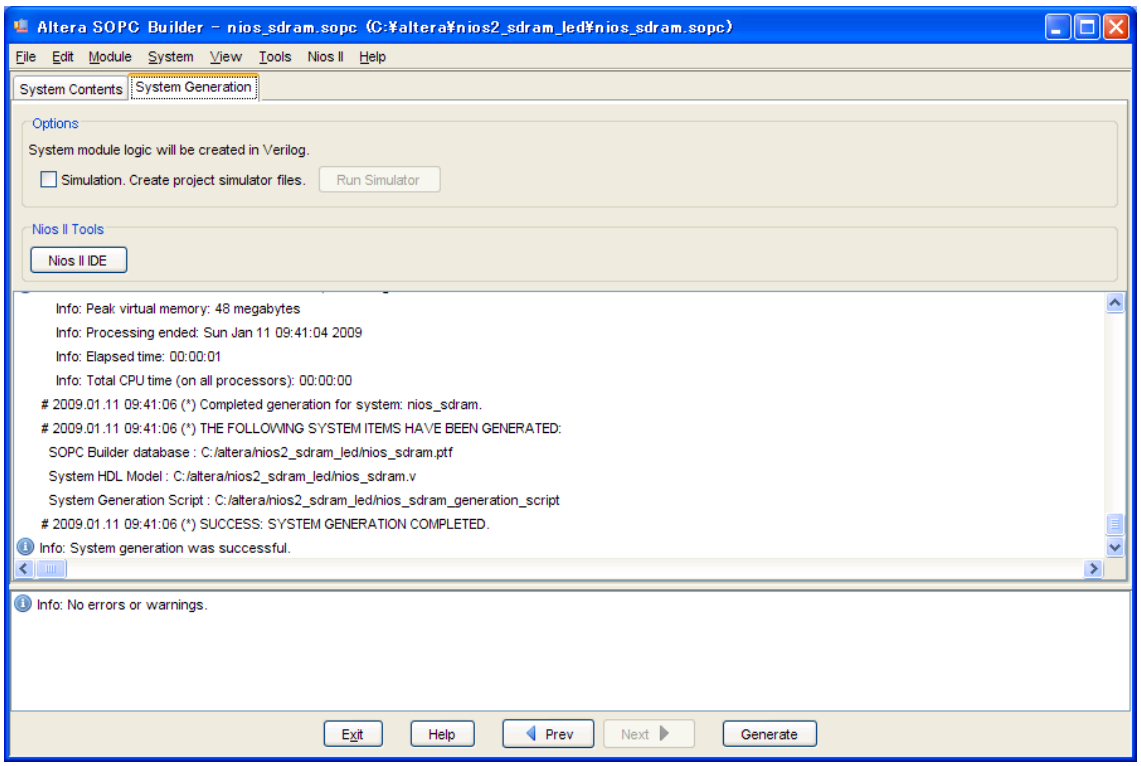
リセットと実行アドレスを SDRAM に設定します。「Finish」を押します。



全部の設定が完了しました。「Generate」ボタンを押して、Nios II システムを生成します。

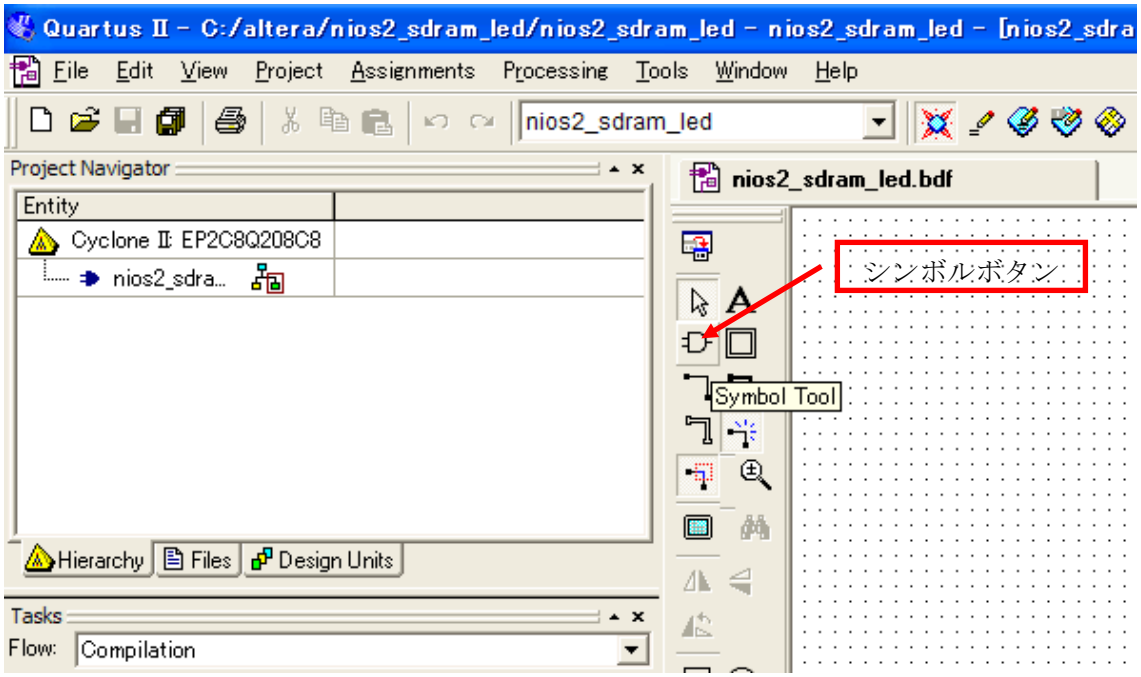


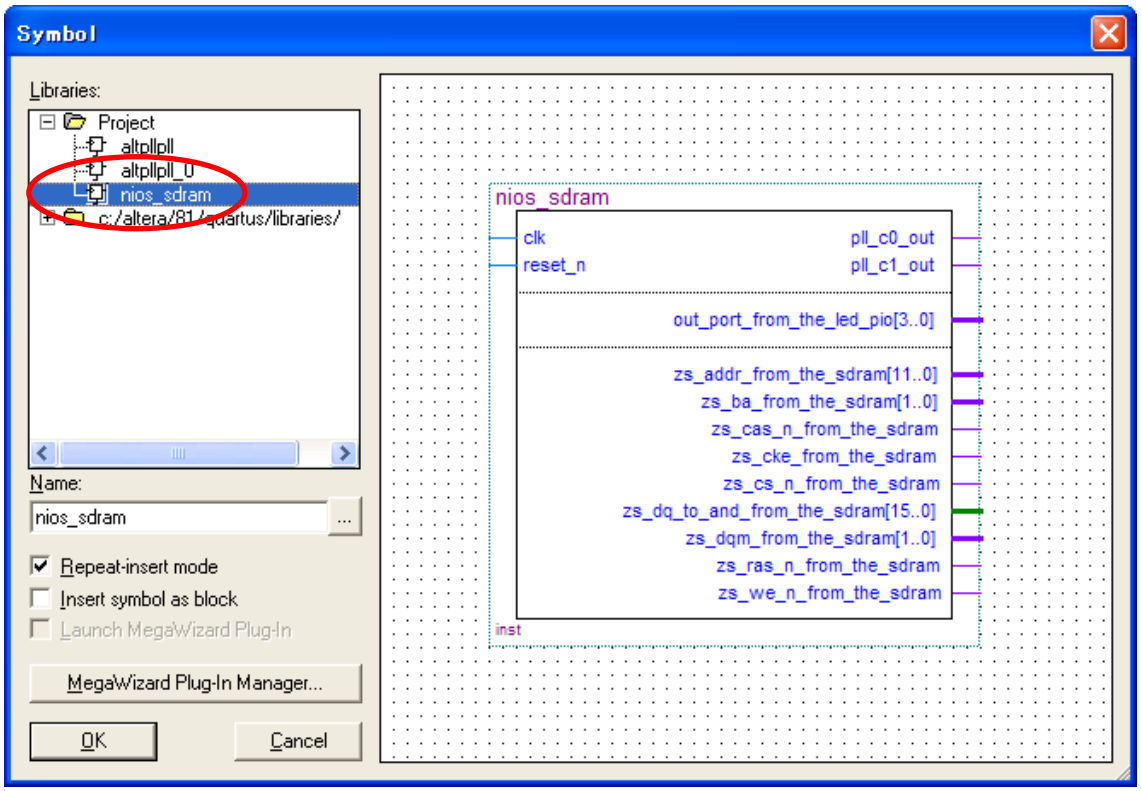
Nios II システムを生成中です。



生成完了すると、「Exit」を押して、Quartus II に戻します。

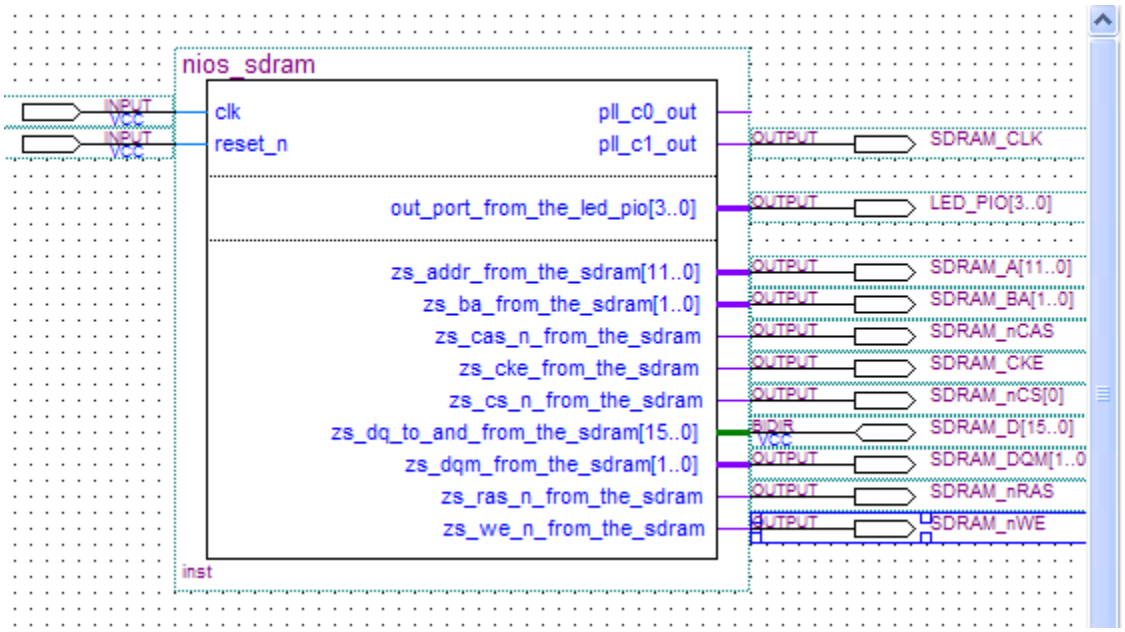
Quartus II で再びシンボルボタンを押すと、





Libraries 欄に、Project 前の[+]記号をクリックすると、中身が表示されます。生成された Nios II システム「nios_sdram」が見えました！nios_sdram を選択して「OK」ボタンを押します。マウス・カーソルに選択されたシンボルがくっついた状態になるので、シンボルを配置したい場所に移動して、マウスの左クリックを押します。配置できた点で ESC キーを押してカーソルを元に戻してください。

次の作業は第四章と同じです。Nios II のシンボルに INPUT/OUTPUT などの端子を配置します。端子の名前は分かりやすい名前に変更してください。



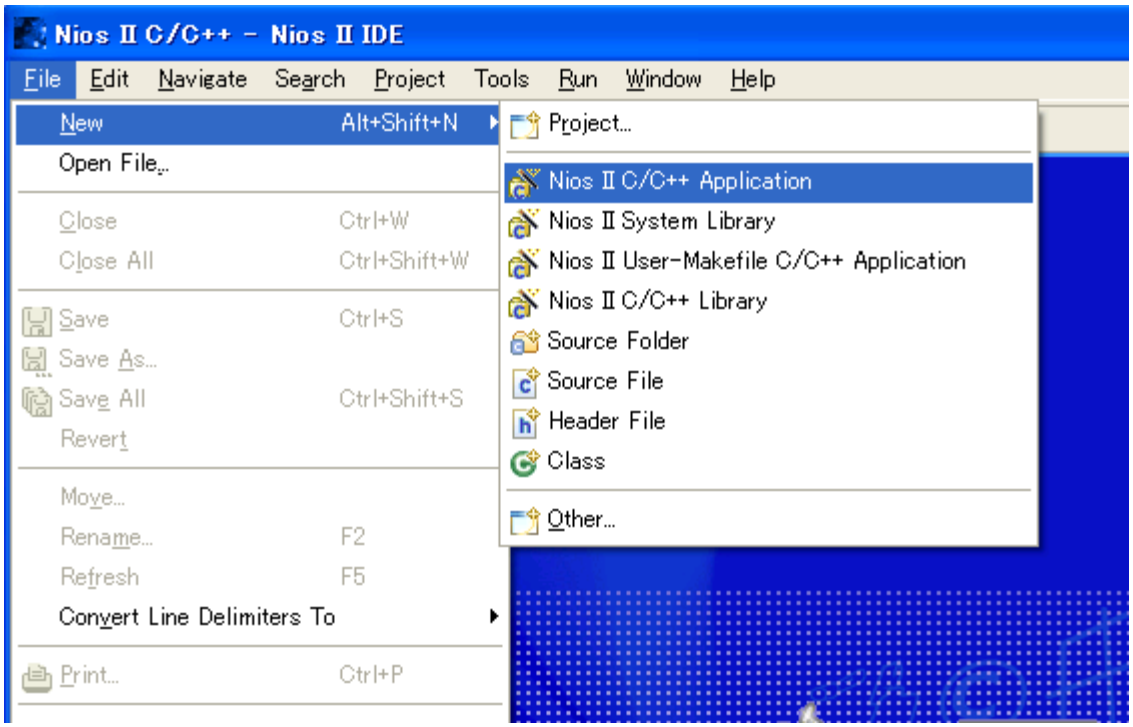
配置が終わったら、コンパイル、ピン・アサイン、再コンパイルを行います。Cyclone II に書き込むデータ*.sof を生成します。このファイルを Cyclone II 書き込みます。

※ EP2C8¥nios_led_sch というフォルダに、すでに完成しているプロジェクト一式があります。

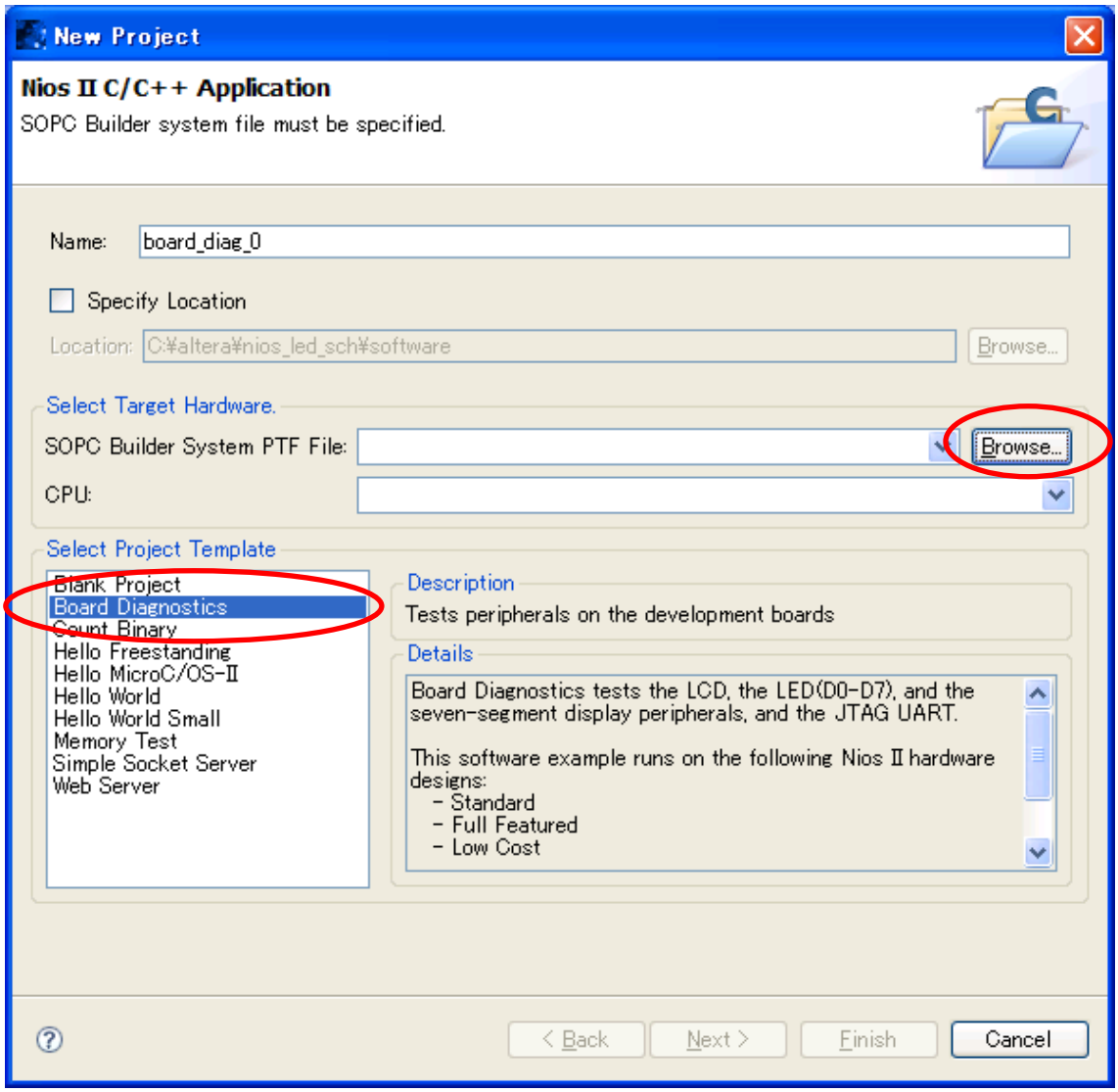
第六章 NIOS II のプログラムの設計

Nios II システムを構築しました。これから、Nios II システムのプログラムを開発します。

Windows の「スタート」→「すべてのプログラム」→「Altera」→「NIOS II EDS 8.1」から NIOS II 8.1 IDE が起動します。

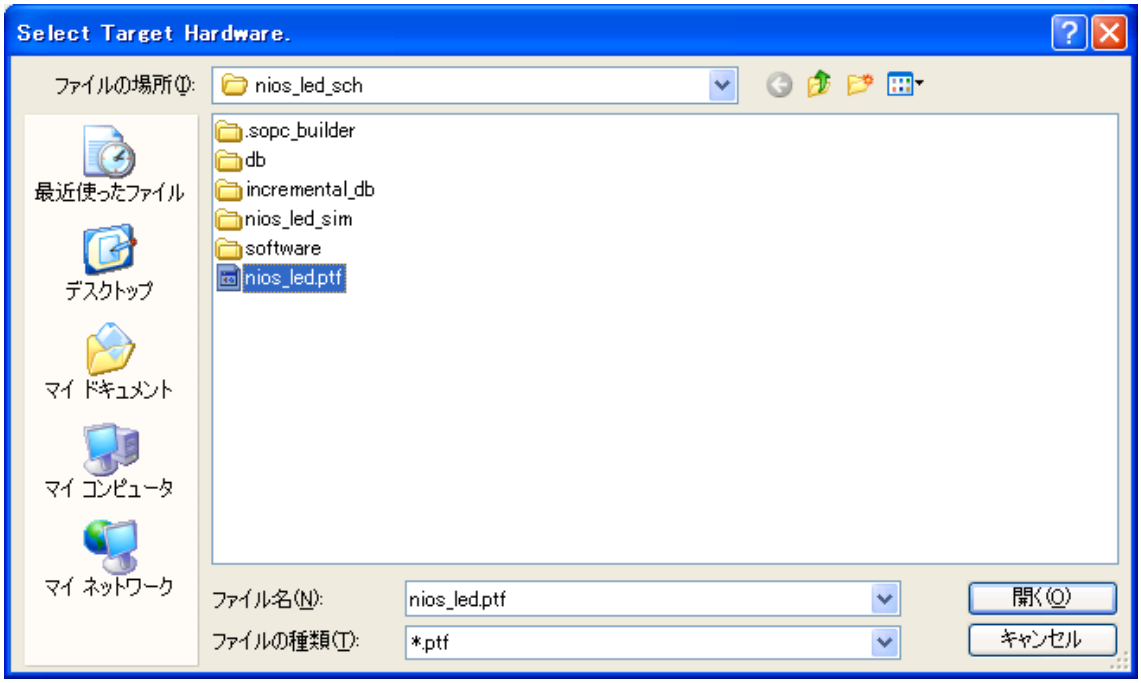


「File」→「New」→「Nios II C/C++ Application」を選択します。

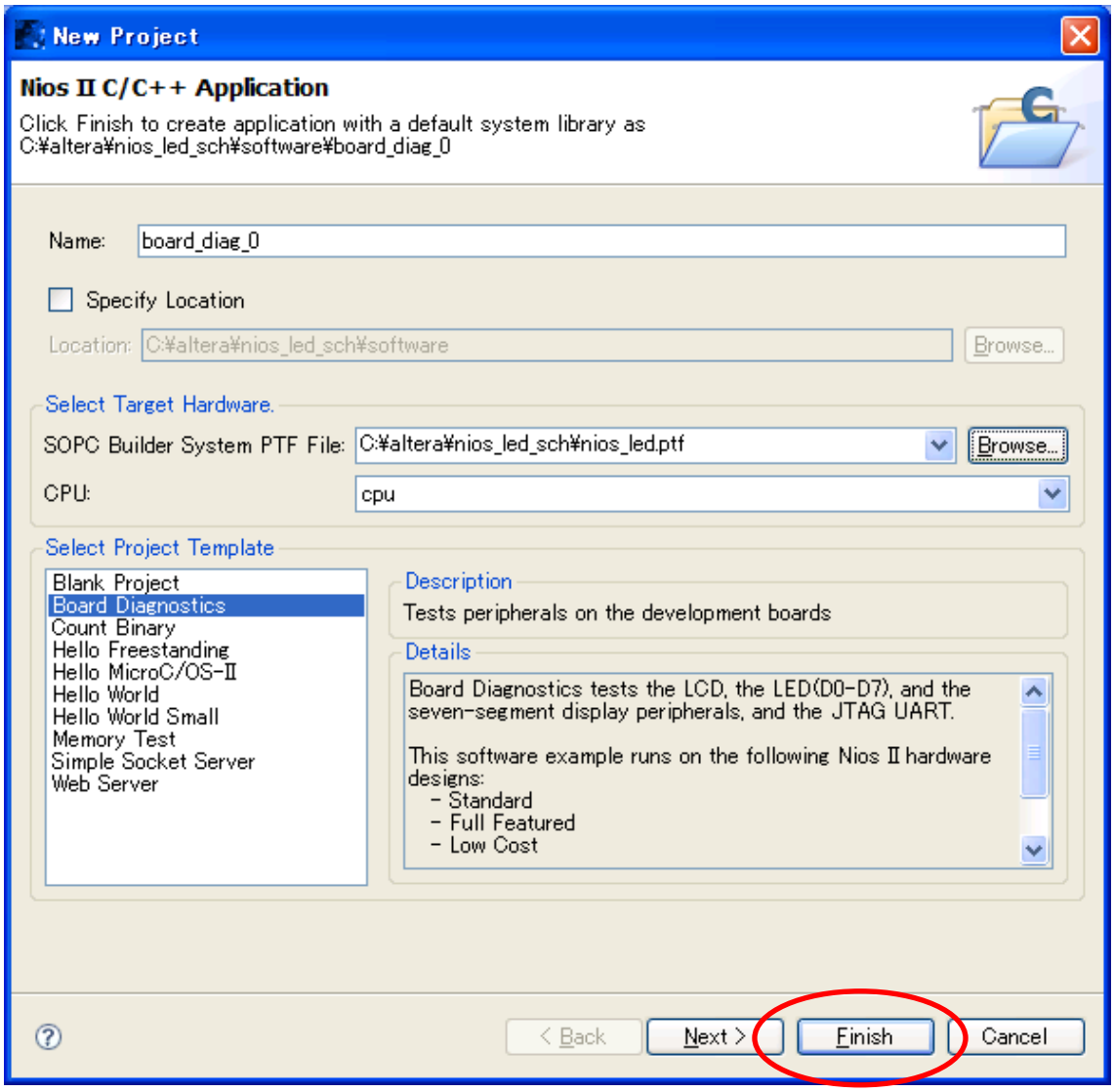


Altera 社は幾つの定番テンプレートを提供しています。これらテンプレートに基づいて、プログラムを開発しやすいです。もちろん、ゼロ「Blank Project」から開発もできます。今回は「Board Diagnostics」を選択します。一行のコードも入力することが必要ないです。

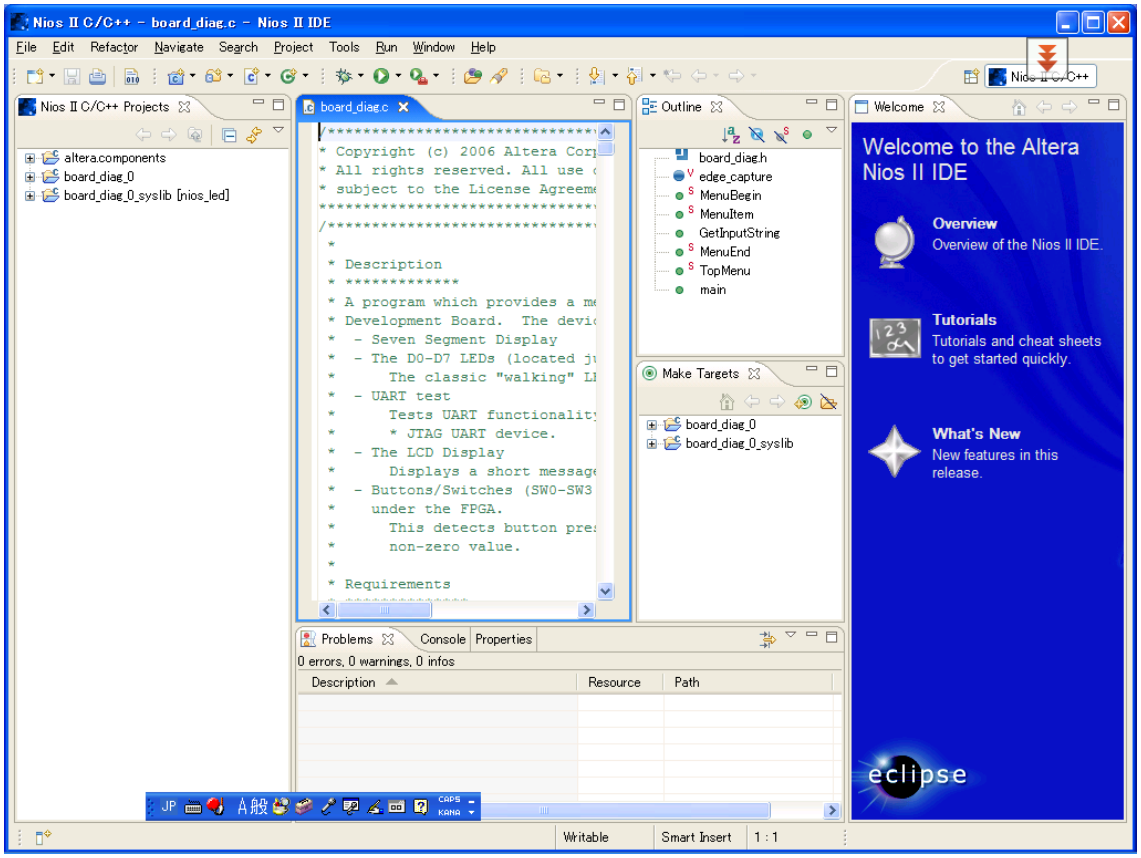
「Browse」ボタンを押して、



Nios II システムの*.ptf ファイルを選択します。



「Finish」ボタンを押すと、プログラムとハードウェアに関連するライブラリを自動的に生成します。



次の手順は「3.5 NIOS II プロセッサーの初体験」で紹介いたしました。