

Nios® II はじめてガイド Nios® II 簡易チュートリアル

Ver.17







Nios® II はじめてガイド Nios® II 簡易チュートリアル

<u>目次</u>

1.	はじめに	3
	ハードウェア	
	2-1. ハードウェア・プロジェクトの作成	
:	2-2. コンフィギュレーション・モードの設定	6
:	2-3. Qsys にてシステムを構成し、HDL を生成	7
;	2-4. Quartus® Prime でコンパイル	12
:	2-5. システムを FPGA ヘダウンロード	15
3.	ソフトウェア	16
:	3-1. Nios® II SBT でソフトウェア・プロジェクトの作成	16
:	3-2. 実行コードをターゲットで実行、デバッグ	20
改	7版履歴	24



1. はじめに

この「Nios®」はじめてガイド」シリーズは、Nios®」プロセッサをはじめて使用するユーザ向けの資料です。

この資料は、インテル[®] Quartus[®] Prime 開発ソフトウェアや Nios[®] II Software Build Tools for Eclipse (Nios[®] II SBT)を使用して、Nios[®] II のシステム生成をチュートリアル形式でまとめたものです。シンプルなハードウェアとソフトウェアを使用し、動作確認を行うまでの最低限必要となる開発フローを理解することができます。

この資料で説明に使用するデザインのハードウェアの構成は、Nios® II コアや JTAG UART、オンチップ・メモリ、 LED 用の8ビット出力用の PIO コンポーネントのみです。ソフトウェアは、LED を点滅させるだけの、非常にシンプルな記述です。

チュートリアルのフローを以下に示します。

【開発ソフトウェア】

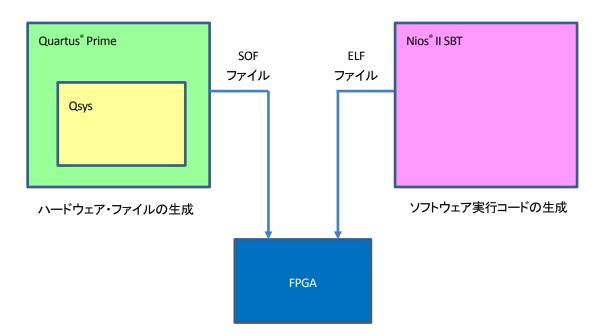
- インテル® Quartus® Prime 開発ソフトウェア 17.0
- Nios® II Software Build Tools for Eclipse (Quartus Prime 17.0) ※ 以降、Nios® II SBT と記載

【フローの概要】

- ① Quartus® Prime の起動
- ② Qsys システム統合ツールにて、Nios® II システムの構築、生成
- ③ Quartus® Prime にてハードウェア・デザインのコンパイル
- ④ ターゲット・ボードへ .sof ファイルをダウンロード
- ⑤ Nios® II SBT にて、ソフトウェア・プロジェクトの作成
- ⑥ ソフトウェア・ソース・ファイルを新規で作成し、記述
- ⑦ ソフトウェア・ソース・ファイルのビルド
- 8 生成されたソフトウェアの実行ファイルをターゲット上で実行
- 9 デバッガを起動し、デバッグ



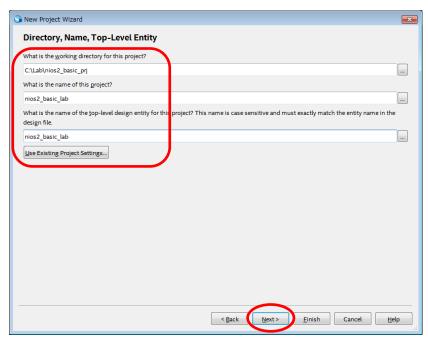
この資料では、Terasic 社の Cyclone® V GX スタータ開発キットを使用して説明します。他の評価ボードなどを使用する場合は、Quartus® Prime にて設定するデバイス型名の選択やピン配置、未使用ピンの処理などをそれぞれのボードの仕様に合わせる必要があります。



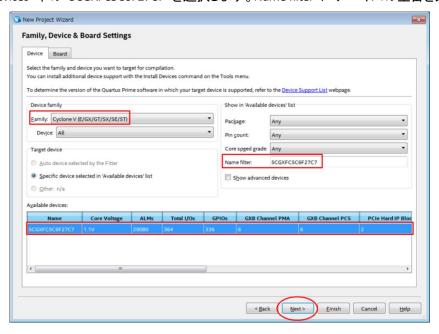


2. ハードウェア

- 2-1. ハードウェア・プロジェクトの作成
- ① Quartus® Prime を起動して、File メニュー ⇒ New Project Wizard を選択します。New Project Wizard の Introduction ウィンドウが表示されたら Next を 1 回クリックして、以下の画面でワーキング・ディレクトリとハードウェア・デザインのプロジェクト名、ハードウェア・デザインのトップのエンティティ名を入力します。



② Next を 3 回クリックし、以下の画面で、ターゲット・デバイスの選択をします。ここでは、Cyclone® V を選択し、 Available devices 中の 5CGXFC5C6F27C7 を選択します。Name filter にデバイスの型番を入力します。

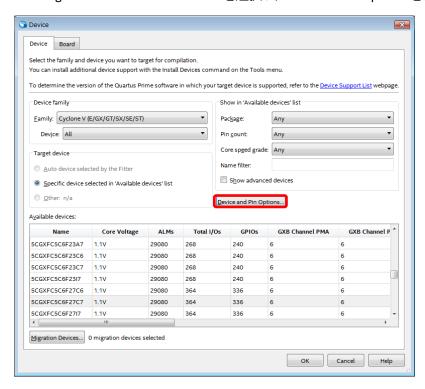


③ Next を 2 回クリックし、表示内容が正しいことを確認して Finish でこのウィンドウを閉じます。この段階で、ハードウェア・プロジェクトが作成されます。

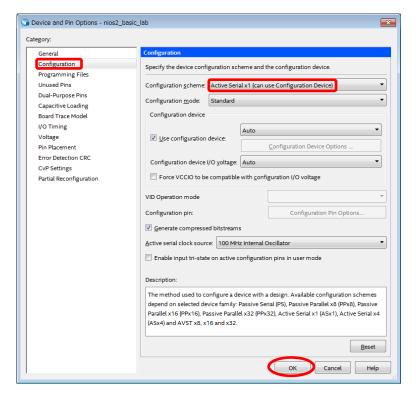


2-2. コンフィギュレーション・モードの設定

① Quartus® Prime の Assignments メニュー ⇒ Device を選択し、Device and Pin Options をクリックします。

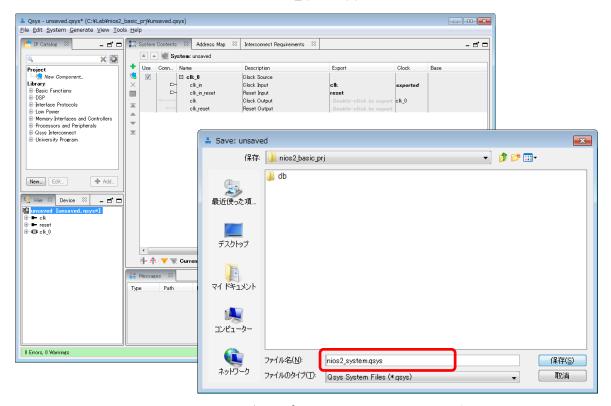


② Device and Pin Options ウィンドウの Configuration を開き、Configuration scheme で Active Serial x1 (もしくは、 Active Serial x4) を選択し、OK をクリックします。

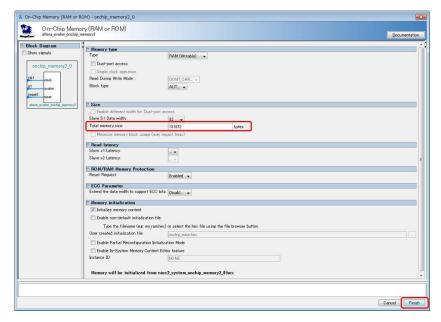




- 2-3. Qsys にてシステムを構成し、HDL を生成
- ① Quartus® Prime の Tools メニュー ⇒ Qsys を選択し、Qsys を開きます。Qsys の File メニュー ⇒ Save as を選択して、先にファイルをセーブします。今回は、nios2_system.qsys というファイル名で保存します。

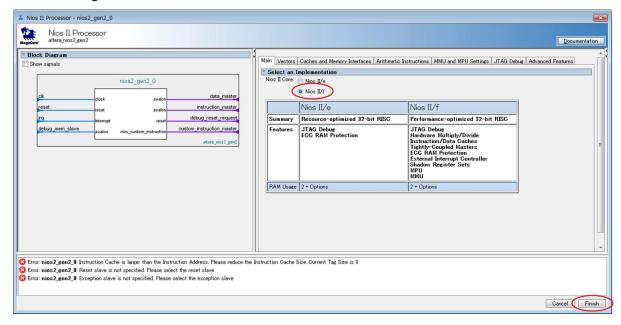


② Qsys 左枠の IP Catalog より Basic Functions グループ内の On-Chip Memory カテゴリにて On-Chip Memory (RAM or ROM) をダブルクリックします。On-Chip Memory (RAM or ROM) ウィンドウ内の Total memory size を 128 K Bytes (128k と入力で反映)に設定し Finish をクリックします。

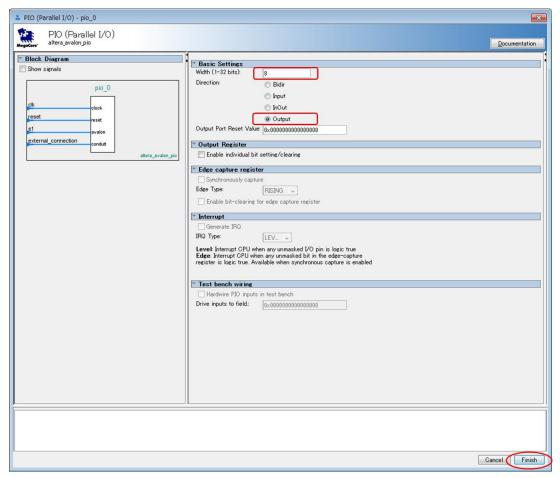




③ Qsys 左枠の IP Catalog より、Processors and Peripherals グループ内の Embedded Processors カテゴリにて Nios II Processor をダブルクリックし Nios II Processor のウィザード内で Nios®II コア を選択しFinish をクリックします。この時点で、Message ウィンドウにエラーが出ますが、気にせず先に進んでください。

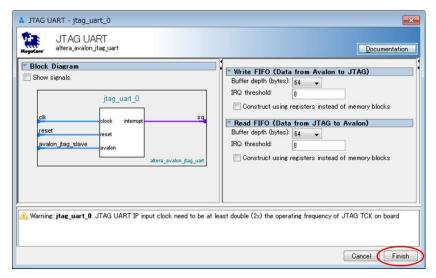


④ Qsys 左枠の IP Catalog より、Processors and Peripherals グループ内の Peripherals カテゴリより PIO(Parallel I/O) をダブルクリックします。 こちらのウィザードはデフォルトの設定で Finish をクリックします。

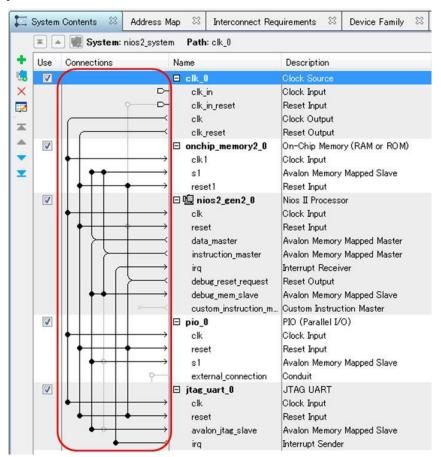




⑤ Qsys 左枠の IP Catalog より、Interface Protocols グループ内の Serial カテゴリより JTAG UART をダブルクリックします。こちらのウィザードもデフォルトの設定で Finish をクリックします。



⑥ 各コンポーネントを接続します。Connections 欄の白丸をクリックすることで下図のように黒丸に変えることで、接続 状態となります。

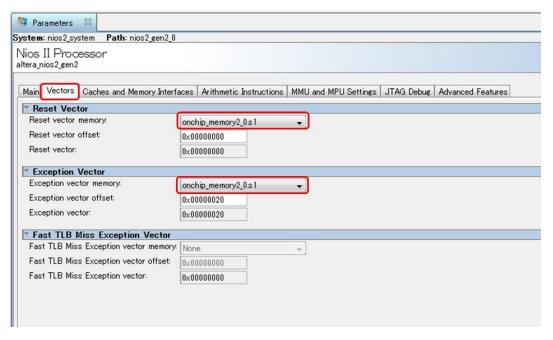




⑦ 割り込みポートを接続します。IRQ 欄の JTAG_UART の Interrupt Sender と Nios II Processor の Interrupt Receiver を下図のように接続します。ここの数字は割り込みの優先順位を表しており、数字が小さいほど優先度が高くなります。今回は割り込みが 1 本なので、0 となっています。



® Nios®II の Reset Vector と Exception Vector の設定を行います。Qsys 上で Nios II Processor をダブルクリックし、 再度 Nios®II の設定画面を開きます。Vectors タブをクリックします。そこで、プルダウン・メニューからオンチップ・メモリを指定します。以下のように設定します。本資料では、Nios®II をオンチップ・メモリからブートする手法を説明します。

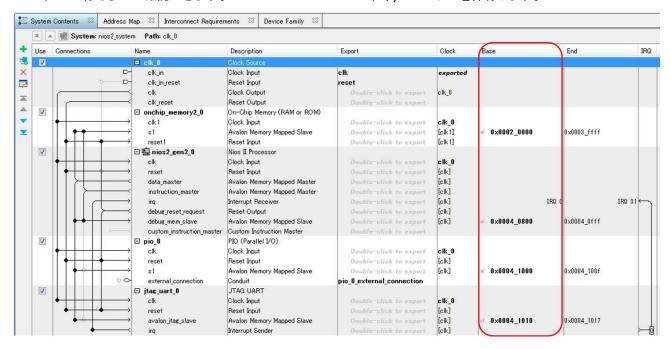


⑨ PIO の出力信号を Qsys の外部に Export します。Export 行にて external_connection に該当するセルをダブル クリックし、Enter で確定させます。

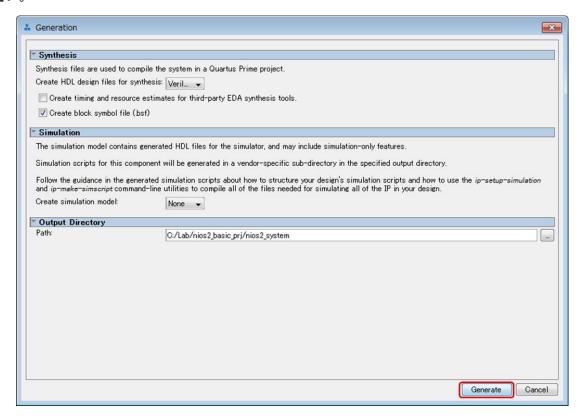




⑪ ベース・アドレスを重複のないように設定します。System メニューの Assign Base Address を実行します。下図のように設定(値は、Quartus® Prime のバージョンや設定順序により、変わる場合があります)され、Message 画面からエラーが消えるのが確認できます。File メニュー ⇒ Save で、Qsys システムを保存します。



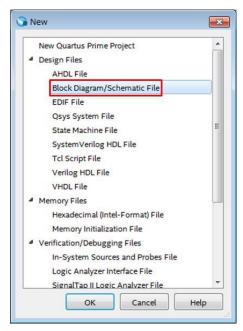
① システムを Generate します。Generate メニューから Generate HDL を選択します。下図の画面が表示されるので、 Generate ボタンをクリックします。Generate が終了したら、Generated Completed 画面の Close をクリックしてく ださい。



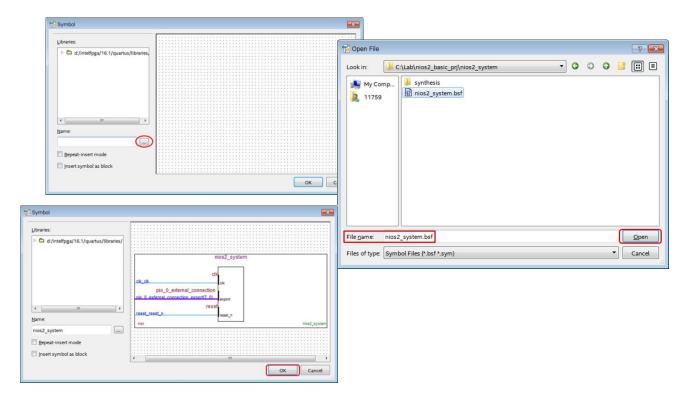


2-4. Quartus® Prime でコンパイル

① 今回はトップのファイルとして回路図(.bdf ファイル)を採用しますが、トップのファイルは HDL でも結構です。
Quartus® Prime の File メニュー ⇒ New を選択し、New ウィンドウより Block Diagram/Schematic File を選択し
OK をクリックします。

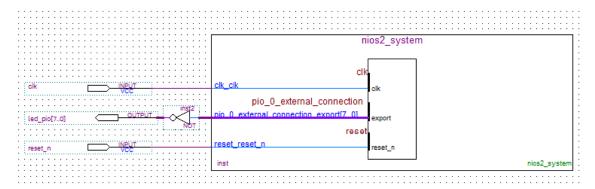


② 表示された Block1.bdf ファイル上でダブルクリックします。Symbol ウィンドウが表示されますので、ファイル選択 ボタンをクリックして、Qsys が生成したフォルダ(nios2_system)内の nios2_system.bsf ファイルを選択し、Open を クリックします。すると、Symbol ウィンドウに該当の BSF ファイルが表示されるので、OK ボタンをクリックして回路 図上に反映させます。

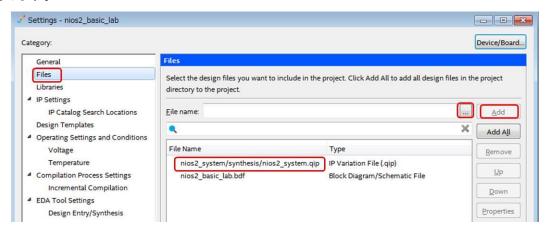




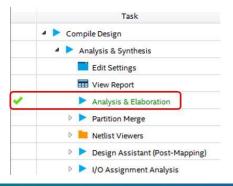
③ 先程と同じように、Block1.bdf ファイル上でダブルクリックします。 Symbol ウィンドウが表示されますので、その Name 欄で input と入力し OK をクリックします。入力ピンのシンボルが用意されるので、Qsys のシンボルに接続します。clk、reset_n に、それぞれ入力ピンのシンボルを接続します。同じように、Name 欄に output と入力することで、出力ピンのシンボルが用意されます。さらに同じように、Name 欄に not と入力することで、インバータのシンボルが用意されるので、led のノードに接続しその外側へ出力ピンのシンボルを接続します。それぞれのシンボルはダブルクリックすることで名前の変更ができます。各シンボルの名前を次の画面の様に、clk、reset_n、led_pio[7..0] としてください。



- ④ File メニュー ⇒ Save As よりトップの .bdf ファイルの保存をします。ここでは、ファイル名を nios2_basic_lab.bdf としました。
- ⑤ Assignment メニュー ⇒ Settings を選択し、Files カテゴリで Qsys が生成した QIP ファイルがプロジェクトに登録されているかを確認します。QIP ファイルがない場合は、ファイル選択ボタンで nios2_system_synthesis/nios2_system.qip を選択し、Add ボタンをクリックします。設定されたら OK ボタンで画面を終了します。



⑥ Quartus® Prime の Processing メニュー ⇒ Start Analysis & Elaboration で、デザインを Quartus® Prime のチェック にかけます。

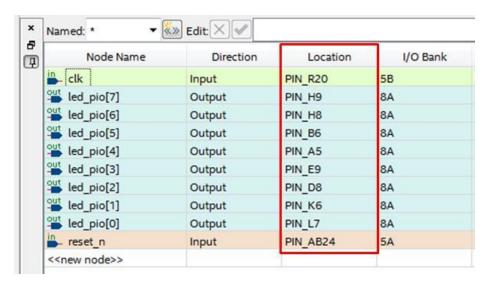




⑦ ピン・アサインを行います。Quartus® Prime の Assignments メニュー ⇒ Pin Planner をクリックします。Pin Planner が表示されるので、次の表に対応するようにピン・アサインを行ってください。Terasic 社の Cyclone V GX スタータ開発キット以外のボードを使用する場合は、使用するボード仕様を確認後、ピン配置の設定をしてください。

clk	R20
led_pio[7]	Н9
led_pio[6]	Н8
led_pio[5]	В6
led_pio[4]	A5
led_pio[3]	E9
led_pio[2]	D8
led_pio[1]	K6
led_pio[0]	L7
reset_n	AB24

⑧ Location 欄へカーソルを持っていき、ピン・アサインを行いたい番号を入力、もしくはプルダウン・メニュー選択してください。

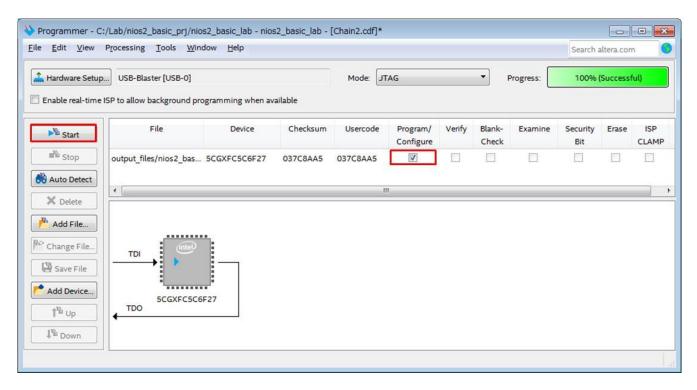


- ⑨ Quartus® Prime の Processing メニュー ⇒ Start Compilation を選択して、トップのデザインをコンパイルします。 ハードウェアの生成は以上です。
 - ※ 本来は、SDC ファイルを生成し、タイミング制約を行う必要がありますが、本資料では省略しています。



2-5. システムを FPGA ヘダウンロード

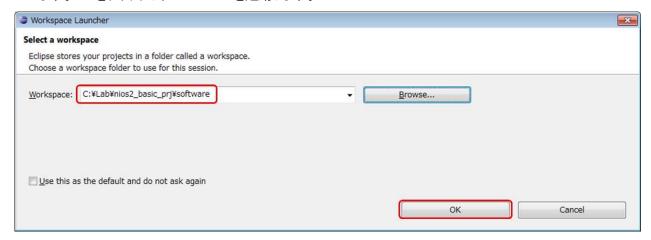
インテル[®] FPGA ダウンロード・ケーブル(USB-Blaster™ もしくは USB-Blaster™ II)を接続し、Quartus[®] Prime の Programmer より SOF ファイルを FPGA にダウンロードします。Tools メニュー ⇒ Programmer を選択します。SOF ファイルを選択し、Program/Configure のチェックを入れて Start をクリックします。





3. ソフトウェア

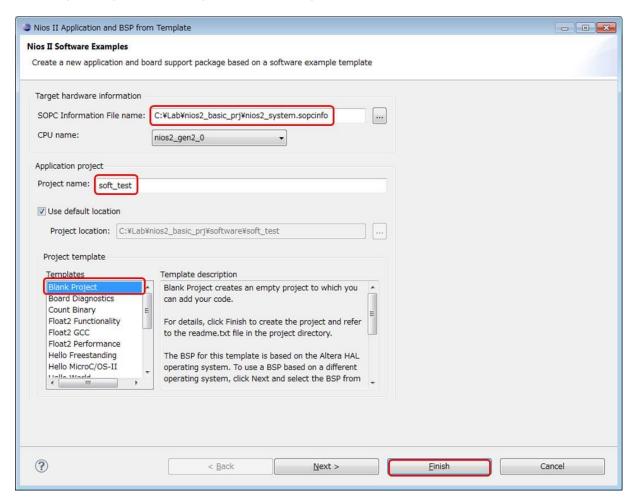
- 3-1. Nios® II SBT でソフトウェア・プロジェクトの作成
- ① Windows® のスタート・メニューより Nios® II SBT を開きます。Workspace Launcher 画面が起動するので、任意のフォルダを指定します。ここでは、予め software という名前のフォルダを作成し、そこをワークスペースとして指定しています。OK をクリックし、Nios® II SBT を起動します。



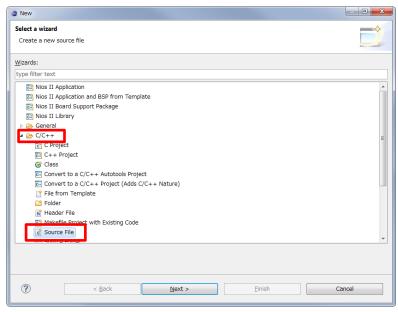
- ② Nios® II SBT でソフトウェアのプロジェクトを作成します。Nios® II SBT の File メニュー ⇒ New ⇒ Nios II Application and BSP from Template をクリックします。Nios II Application and BSP from Template 画面が起動します。
- ③ SOPC Information File name に、SOPCINFO ファイルを選択します。これは、Qsys 生成時に作成されたファイルで、Qsys 内のシステム情報をソフトウェアに渡すために参照されるファイルとなります。
- ④ Project name に任意のプロジェクト名を入力します。ここでは、soft_test と入力します。



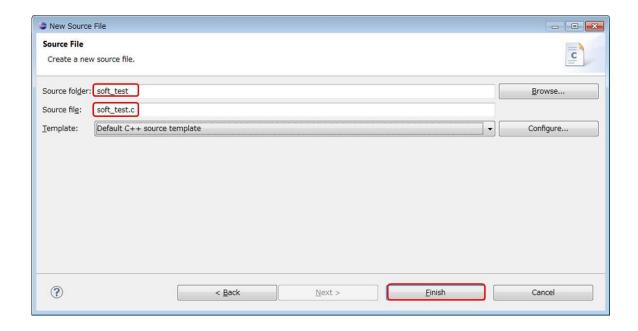
⑤ 最後に Project Template 欄の Templates で、Blank Project を選択し、Finish をクリックします。



⑥ Nios®II のソフトウェアを記述します。Nios®II SBT の File メニュー ⇒ New ⇒ Other ⇒ C/C++ ⇒ Source File を選択し、Next をクリックします。New Source File ウィンドウの、Source Folder 欄に、Browse ボタンよりソフトウェア・プロジェクトのアプリケーション・プロジェクト(_bsp がついていない方のフォルダ)を選択します。Source File 欄には、任意のソフトウェアのファイル名を拡張子.c をつけて入力し、Finish をクリックします。ここでは、soft_test.c と入力します。







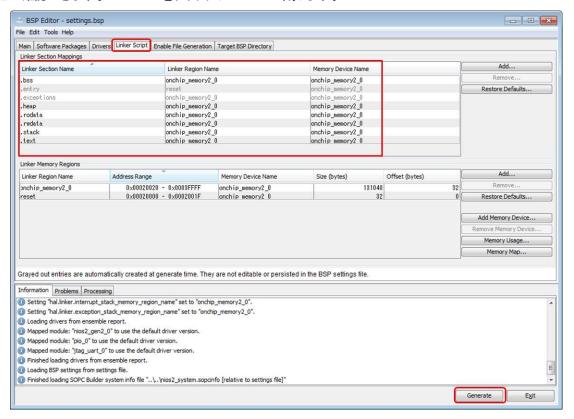
⑦ Nios® II SBT に新規でファイルが用意されたので、そちらにソフトウェアのコードを記述します。以下の記述を行ってください。

```
#include <stdio.h>
#include unistd.h>
#include "system.h"
#include "altera_avalon_pio_regs.h"

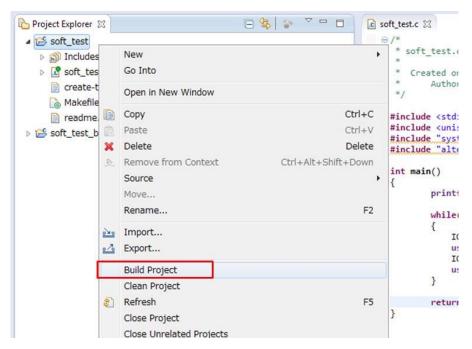
int main()
{
    printf( "Hello from Nios II! Start!\n");
    while(1)
    {
        IOWR_ALTERA_AVALON_PIO_DATA(PIO_0_BASE, 0x55);
        usleep(500000);
        IOWR_ALTERA_AVALON_PIO_DATA(PIO_0_BASE, 0xAA);
        usleep(500000);
    }
    return 0;
}
```



⑧ システムの設定を確認します。soft_test_bsp フォルダを右クリックし、NiosⅡ ⇒ BSP Editor を実行すると NiosⅡ BSP Editor が起動します。Linker Script タブを指定し、プログラム・メモリに、 onchip_memory2_0 が設定されていることが確認できます。Generate をクリックして Exit で閉じます。

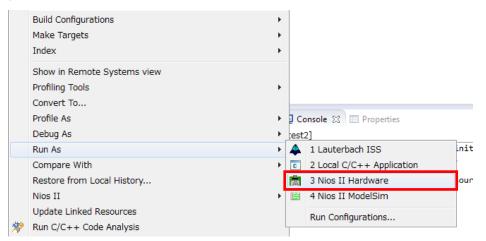


⑨ ソフトウェアをビルドします。Nios®II SBT 左枠のアプリケーション・プロジェクトのフォルダ(_bsp がついていない方)をハイライトし、右クリック ⇒ Build Project をクリックします。





- 3-2. 実行コードをターゲットで実行、デバッグ
- ① ビルドしたソフトウェアをターゲットのシステムで実行します。Nios®II SBT 左枠のアプリケーション・プロジェクトのフォルダ(_bsp がついていない方)をハイライトし、右クリック ⇒ Run As ⇒ Nios II Hardware をクリックします。



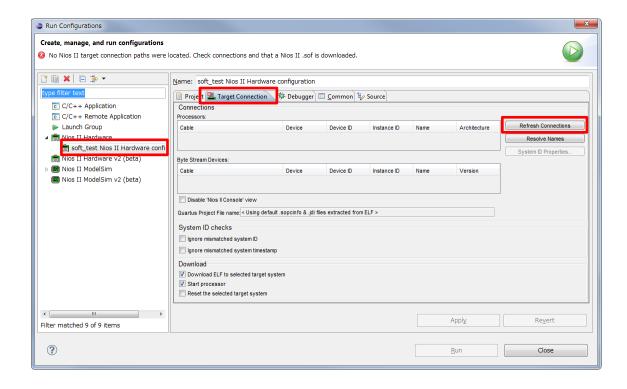
② Nios®Il Console ウィンドウに、printf の出力のキャラクタが確認できます。また、ターゲット上の LED の点灯がプログラムの通りの動作になっていることを確認します。



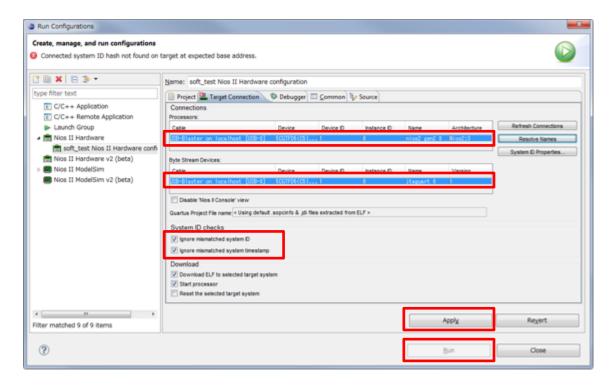
上図のように「Hello from Nios II! Start!」とキャラクタが出力(表示)されたら、③へ進みます。



上図のようなキャラクタが出力(表示)されなくて Run Configurations が立ち上がった場合、Target Connection タブを選択して、右上の Refresh Connections ボタンをクリックします。

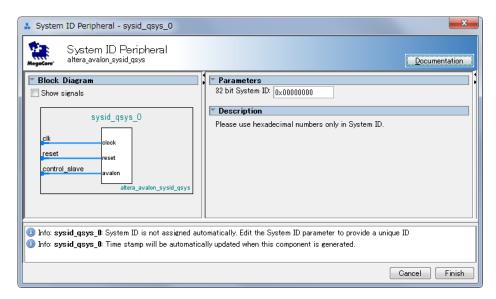


下図のようにインテル FPGA ダウンロード・ケーブル(USB-Blaster™ など)を検出してターゲット・ボードとの接続が確認できたら、Ignore mismatched system ID と Ignore mismatched system timestamp の両方にチェックを入れて、右下にある Apply ボタンをクリック後に、その下の Run ボタンもクリックします。

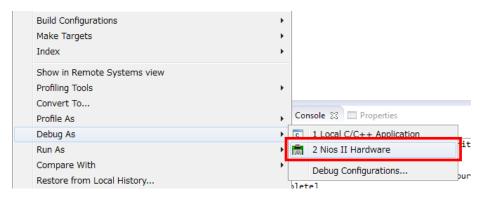




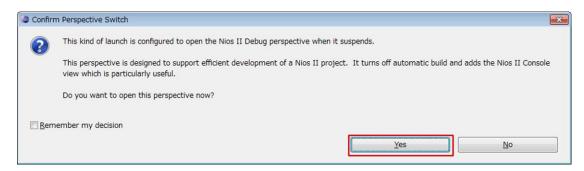
この資料の説明では、Qsys システムに System ID Peripheral を追加していませんが、事前に追加しておくとこの接続できないエラーが回避できます。



③ デバッガを起動します。Nios® II SBT 左枠のアプリケーション・プロジェクトのフォルダ(_bsp がついていない方)をハイライトし、右クリック ⇒ Debug As ⇒ Nios II Hardware をクリックします。

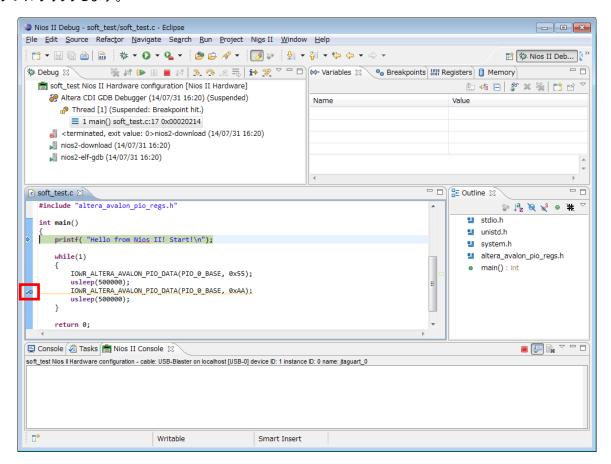


④ デバッグ・ウィンドウを開くために以下のウィンドウで、Yes をクリックします。

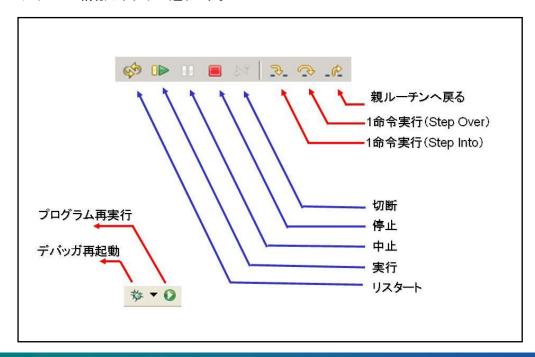




⑤ Nios®IISBT のウィンドウがデバッグ・ウィンドウに切り替わり、デバッグが行えるようになりました。ここで、ソフトウェア・ブレークポイントやステップ実行等が行うことができます。ブレークポイントは、プログラム・ソースコードが表示されているウィンドウの左端をダブルクリックすることにより設定します。ブレークポイントを設定した箇所にはアイコンが表示されます(下画面赤枠をご参照ください)。設定を解除する場合は、このブレークポイントのアイコンをダブルクリックします。



⑥ デバッガのアイコンの情報は、以下の通りです。





改版履歴

Revision	年月	概要
1	2017年11月	初版

免責およびご利用上の注意

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

- 1. 本資料は非売品です。許可無く転売することや無断複製することを禁じます。
- 2. 本資料は予告なく変更することがあります。
- 3. 本資料の作成には万全を期していますが、万一ご不明な点や誤り、記載漏れなどお気づきの点がありましたら、本資料を入手されました下記代理店までご一報いただければ幸いです。 株式会社マクニカ アルティマ カンパニー https://www.alt.macnica.co.jp/ 技術情報サイト アルティマ技術データベース https://www.altma.jp/members/

株式会社エルセナ http://www.elsena.co.jp 技術情報サイト ETS https://www.elsena.co.jp/elspear/members/index.cfm

- 4. 本資料で取り扱っている回路、技術、プログラムに関して運用した結果の影響については、責任を負いかねますのであらかじめご了承ください。
- 5. 本資料は製品を利用する際の補助的な資料です。製品をご使用になる際は、各メーカ発行の英語版の資料もあわせてご利用ください。