

# Nios<sup>®</sup> II はじめてガイド Nios<sup>®</sup> II 簡易シミュレーション

Ver.17

# Nios® II はじめてガイド

## Nios® II 簡易シミュレーション

### 目次

1. はじめに .....	3
2. 操作方法 .....	4
2-1. Qsys の設定 .....	4
2-2. Nios® II SBT の設定 .....	6
2-3. シミュレーションの実行 .....	9
改版履歴 .....	11

## 1. はじめに

この「Nios® II はじめてガイド」シリーズは、Nios® II 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、別資料「Nios® II はじめてガイド - Nios® II 簡易チュートリアル」で作成した Nios® II システムのシミュレーションを、ModelSim® - Intel® FPGA Edition を使って行うための操作マニュアルです。

この資料では ModelSim® - Intel® FPGA Edition を使って説明していますが、ModelSim® などの他のシミュレータでも同様に実行できます。

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
- ModelSim® - Intel® FPGA Edition 10.5b (Quartus® Prime 17.0)
- Nios® II Software Build Tools for Eclipse (Quartus® Prime 17.0) ※以降、Nios® II SBT と記載

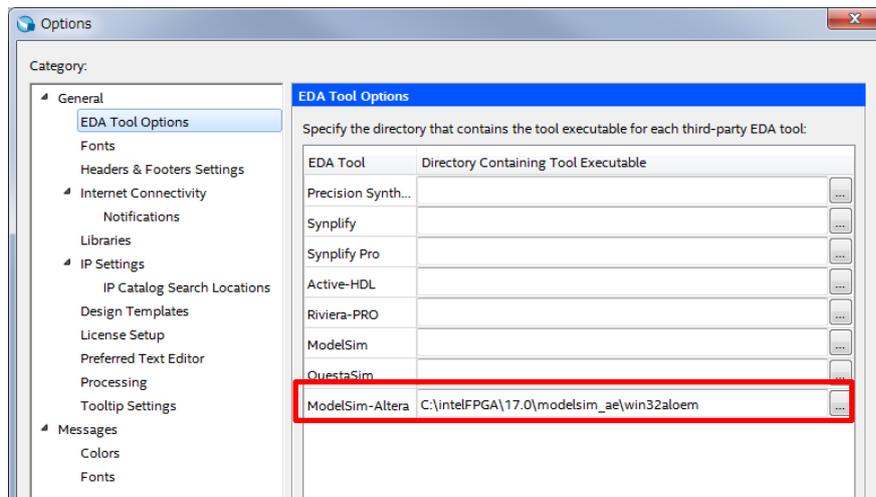
### 【フローの概要】

- ① Qsys システム統合ツールで、テストベンチなどのシミュレーションに必要なファイルを生成
- ② Nios® II SBT のソフトウェア・プロジェクトを開き、シミュレーション用の最適化オプションを設定して、ビルドを実行
- ③ Nios® II SBT から、シミュレーションを実行
- ④ 起動した ModelSim® - Intel® FPGA Edition で、波形表示のためのコマンドを実行
- ⑤ シミュレーション実行のコマンドを入力して、シミュレーションを実行
- ⑥ シミュレーション結果を確認

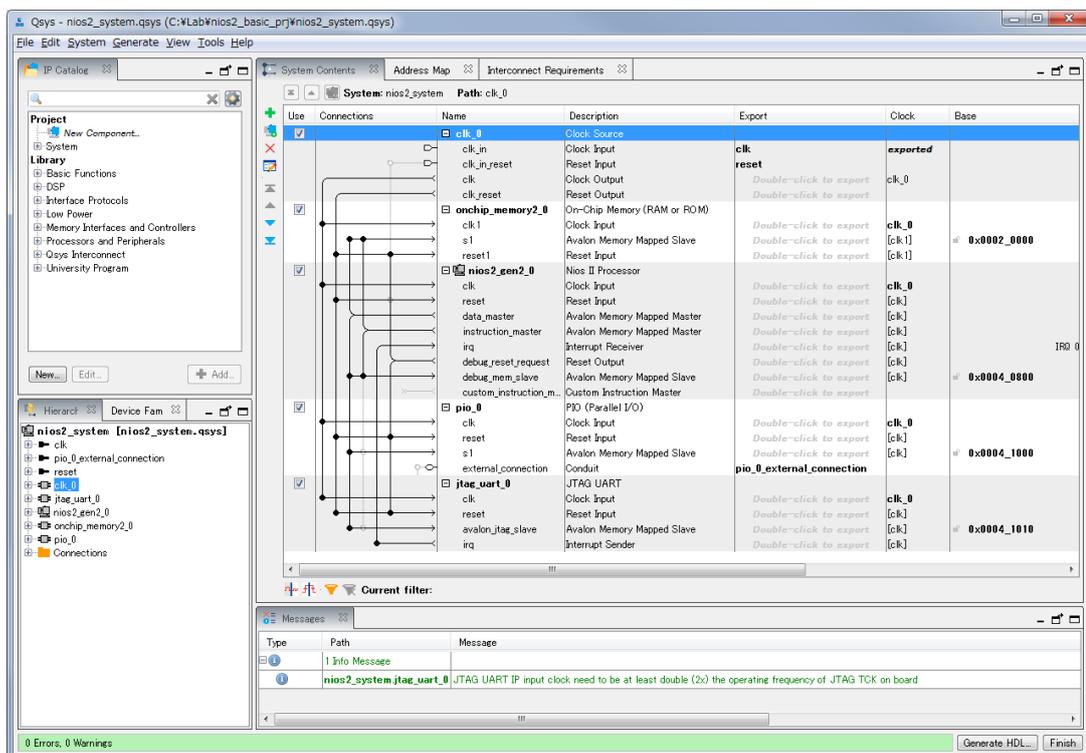
## 2. 操作方法

### 2-1. Qsys の設定

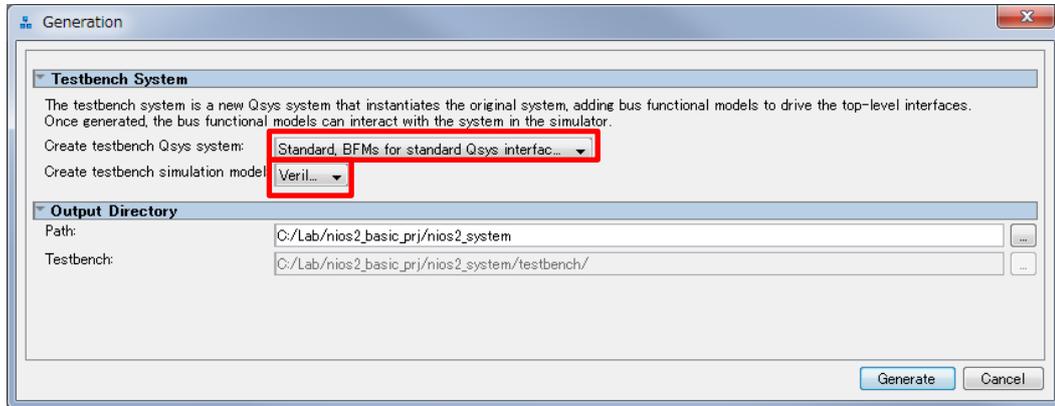
- ① 別資料「Nios® II はじめてガイド - Nios® II 簡易チュートリアル」では、C:\Lab\nios2\_basic\_prj フォルダに nios2\_basic\_lab というプロジェクトを用意しました。この資料でも、同じフォルダ構成を例として進めていきます。
- ② Quartus® Prime を起動し、File メニュー ⇒ Open Project から nios2\_basic\_prj フォルダ内の nios2\_basic\_lab.qpf ファイルを選択し、プロジェクトを開きます。
- ③ 使用する ModelSim® - Intel® FPGA Edition のパスを確認します。Quartus® Prime の Tools メニュー ⇒ Options を選択し、EDA Tool Options カテゴリを確認します。ModelSim-Altera 欄に ModelSim® - Intel® FPGA Edition の実行ファイルが格納されているフォルダのパスが設定されていることを確認します。



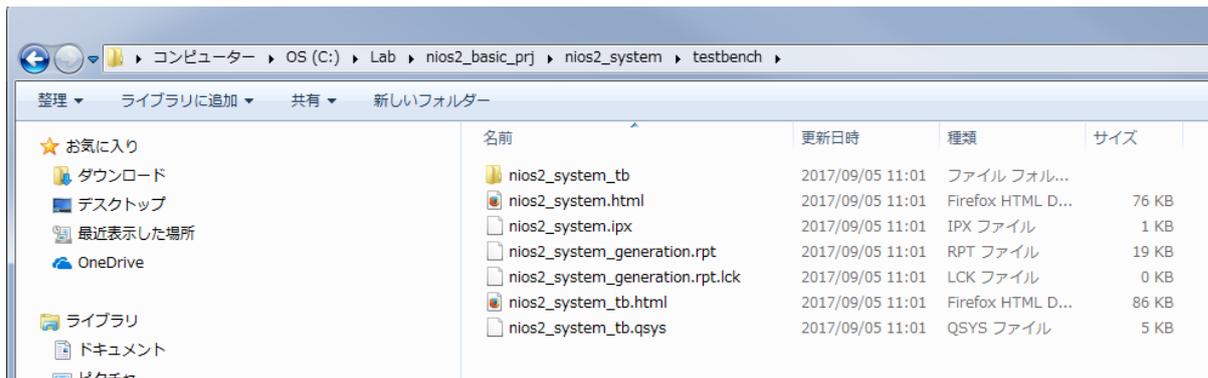
- ④ nios2\_basic\_lab のプロジェクトで Qsys を起動します。Quartus® Prime の Tools メニュー ⇒ Qsys を選択し、nios2\_system.qsys を選択します。



- ⑤ Qsys で Testbench System を Generate します。Generate メニュー ⇒ Generate Testbench System を実行します。Generation 画面が起動するので、以下のように設定して Generate ボタンをクリックします。Testbench で指定されているパスに、シミュレーションに必要なファイル一式が生成されます。



- ⑥ Generate が終了したら、Generate Completed 画面を Close で終了します。  
 ⑦ 生成された testbench フォルダの中には、以下のように各種ファイルが生成されていることが確認できます。



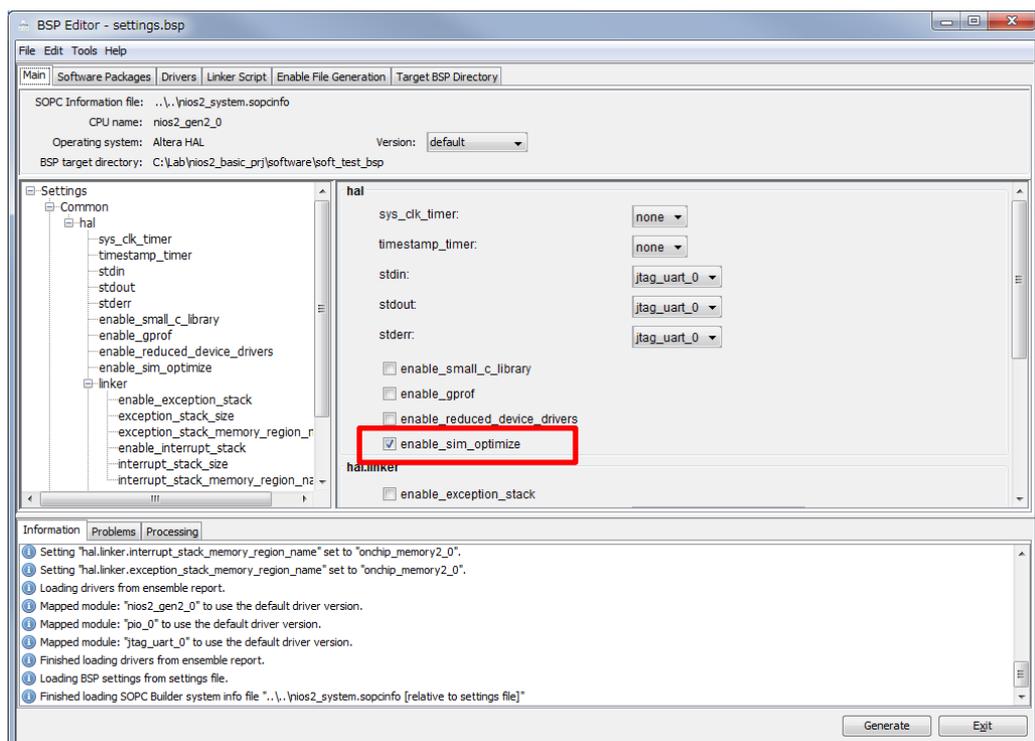
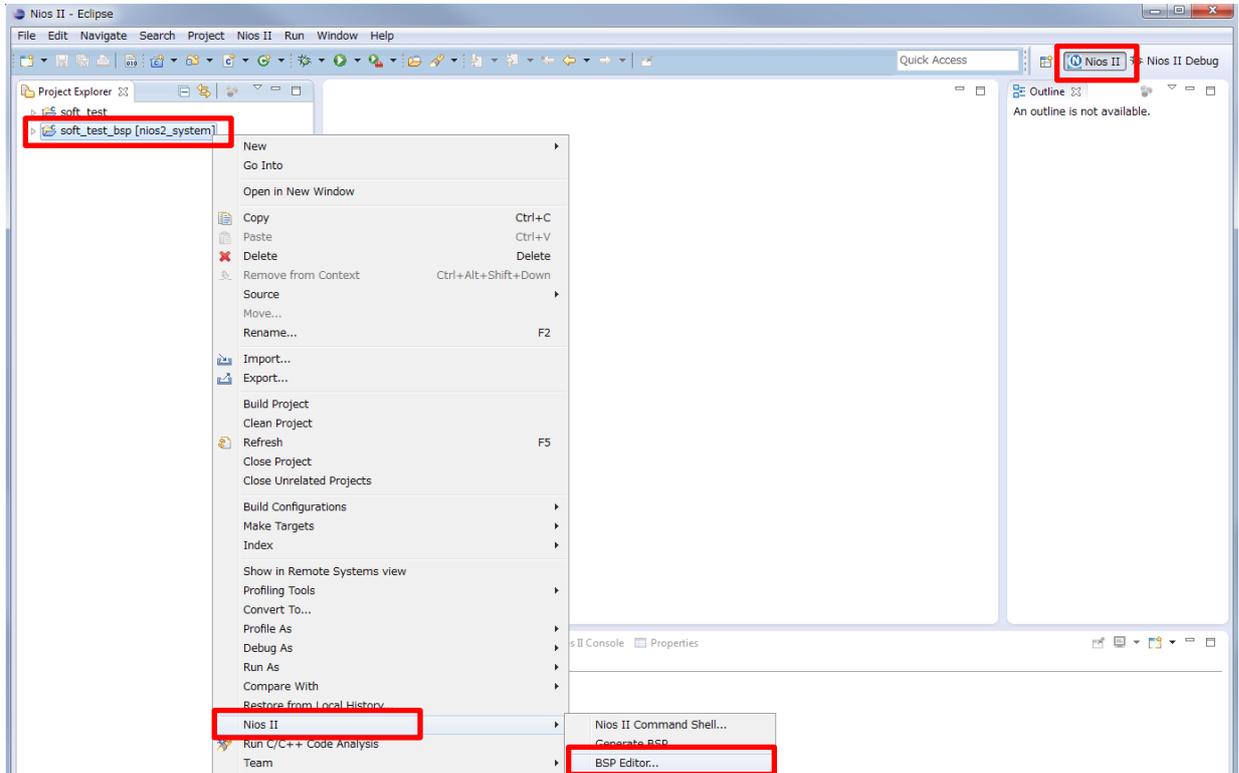
- ⑧ testbench フォルダ以下の nios2\_system\_tb¥simulation¥nios2\_system\_tb.v を見るとテストベンチ・ファイルが確認できます。Qsys のシステムとクロック、リセット生成用のインスタンス、PIO 出力用のインスタンスが確認できます。

```
// nios2_system_tb.v
↓
// Generated using ACDS version 17.0 5954
↓
timescale 1 ps / 1 ps
module nios2_system_tb (
);
↓
wire nios2_system_inst_clk_bfm_clk_clk; // nios2_system_inst_clk_bfm:clk -> [nios2_system_inst:clk_clk, nios2_system_inst_reset_bfm:clk]
wire [7:0] nios2_system_inst_pio_0_external_connection_export; // nios2_system_inst_pio_0_external_connection_export -> nios2_system_inst_pio_0_external_connection_bfm:sig_export
wire nios2_system_inst_reset_bfm_reset_reset; // nios2_system_inst_reset_bfm:reset -> nios2_system_inst.reset_reset_n

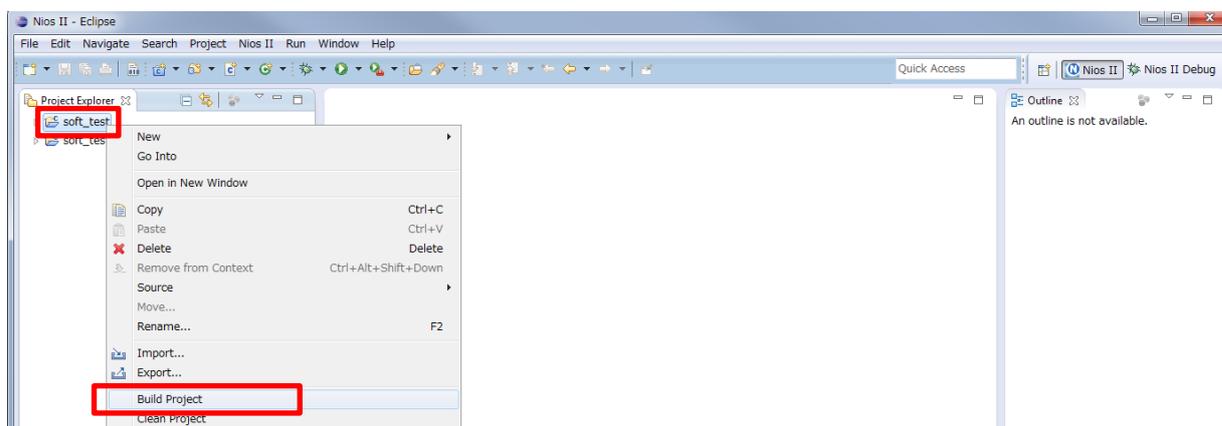
nios2_system nios2_system_inst (
    .clk_clk (nios2_system_inst_clk_bfm_clk_clk), // clk_clk
    .pio_0_external_connection_export (nios2_system_inst_pio_0_external_connection_export), // pio_0_external_connection.export
    .reset_reset_n (nios2_system_inst_reset_bfm_reset_reset) // reset.reset_n
);
↓
altera_avalon_clock_source #(
    .CLOCK_RATE (50000000),
    .CLOCK_UNIT (1)
) nios2_system_inst_clk_bfm (
    .clk (nios2_system_inst_clk_bfm_clk_clk) // clk_clk
);
↓
altera_conduit_bfm nios2_system_inst_pio_0_external_connection_bfm (
    .sig_export (nios2_system_inst_pio_0_external_connection_export) // conduit.export
);
↓
endmodule
```

## 2-2. Nios® II SBT の設定

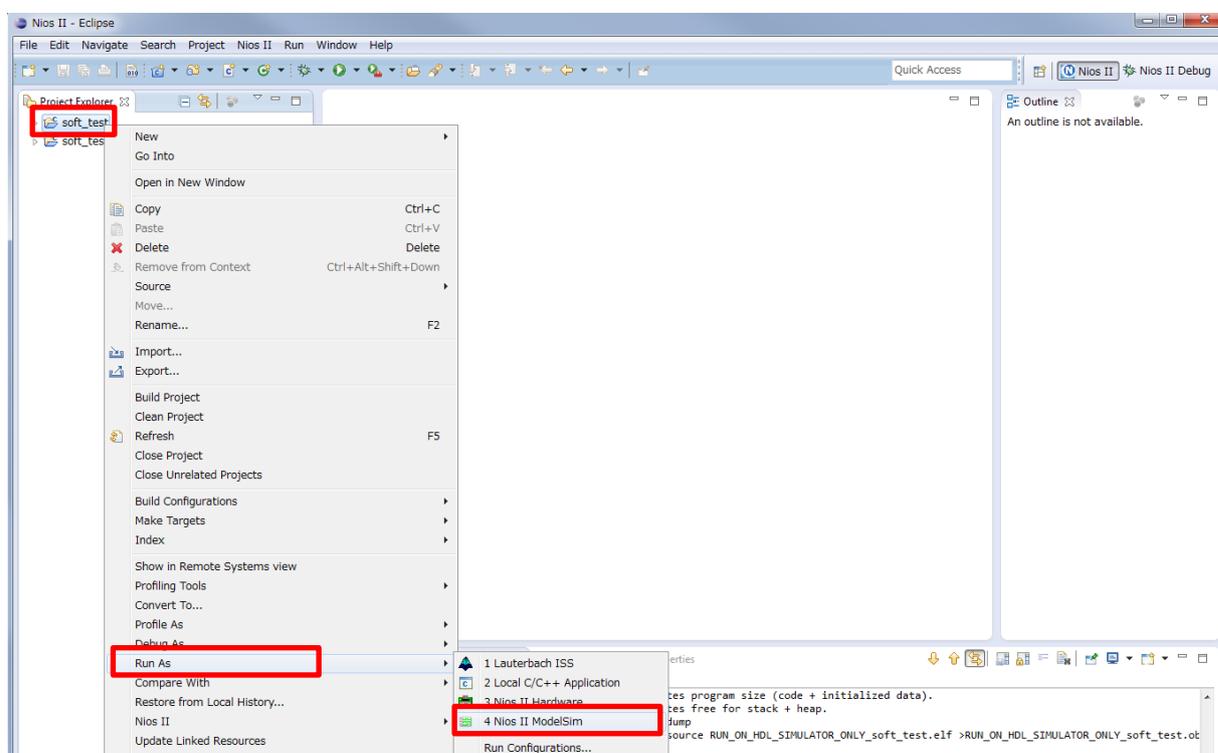
- ① 別資料「Nios® II はじめてガイド - Nios® II 簡易チュートリアル」で作成した Nios® II SBT のソフトウェアプロジェクト soft\_test を開きます。
- ② BSP Editor を起動し、シミュレーション用の最適化設定を行います。soft\_test\_bsp を右クリック ⇒ Nios II ⇒ BSP Editor を実行します。下図のように、enable\_sim\_optimize にチェックを入れます。この設定を行うことで、シミュレーション時間を短縮します。Generate ⇒ Exit の順でクリックして、設定を反映させます。



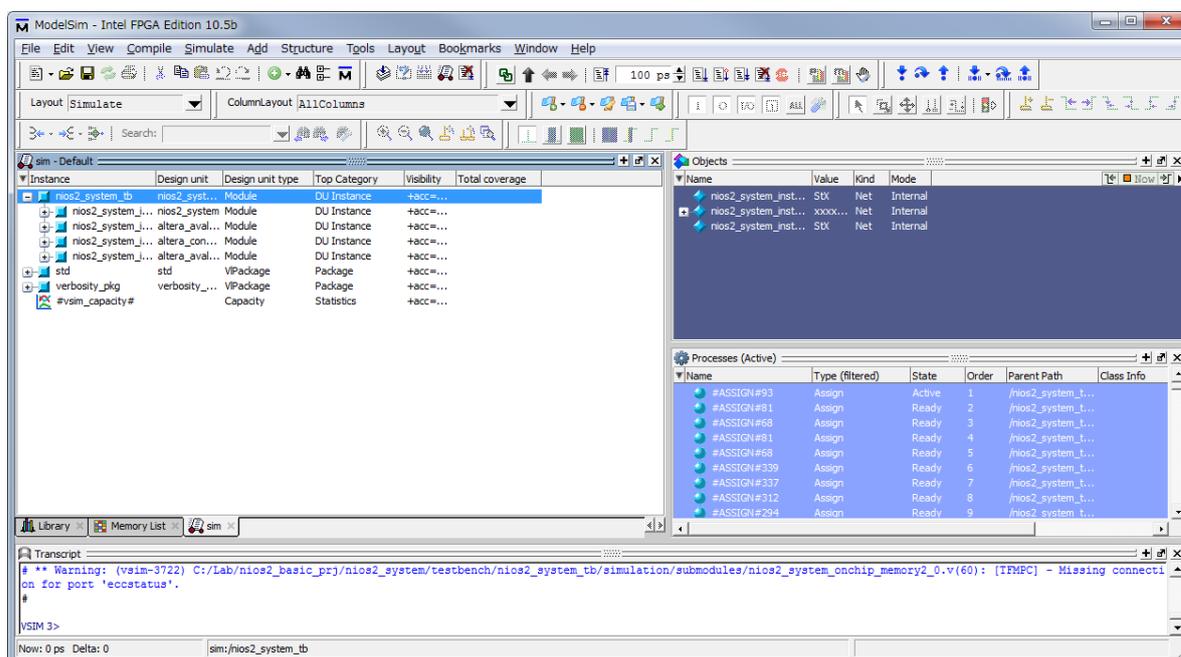
- ③ プロジェクトのビルドを実行します。soft\_test を右クリック ⇒ Build Project を実行します。



- ④ シミュレーションを実行します。soft\_test を右クリック ⇒ Run as ⇒ Nios II ModelSim を実行します。



- ⑤ ModelSim® が起動し、ライブラリの生成や必要なファイルのコンパイルとロードを自動的に行います。



- ⑥ この例では、ModelSim® - Intel® FPGA Edition が以下のフォルダで実行されています。

```
C:\Lab\mios2_basic_prj\software\soft_test\obj\default\runtime\sim\mentor
```

コンソール画面でキーボードの ↑ を 1 回押すと、1 つ前に実行したコマンド(履歴)が表示されます。以下のコマンドを自動実行していることが確認できます。

```
VSIM 1>do msim_setup.tcl
```

```
VSIM 2>ld
```

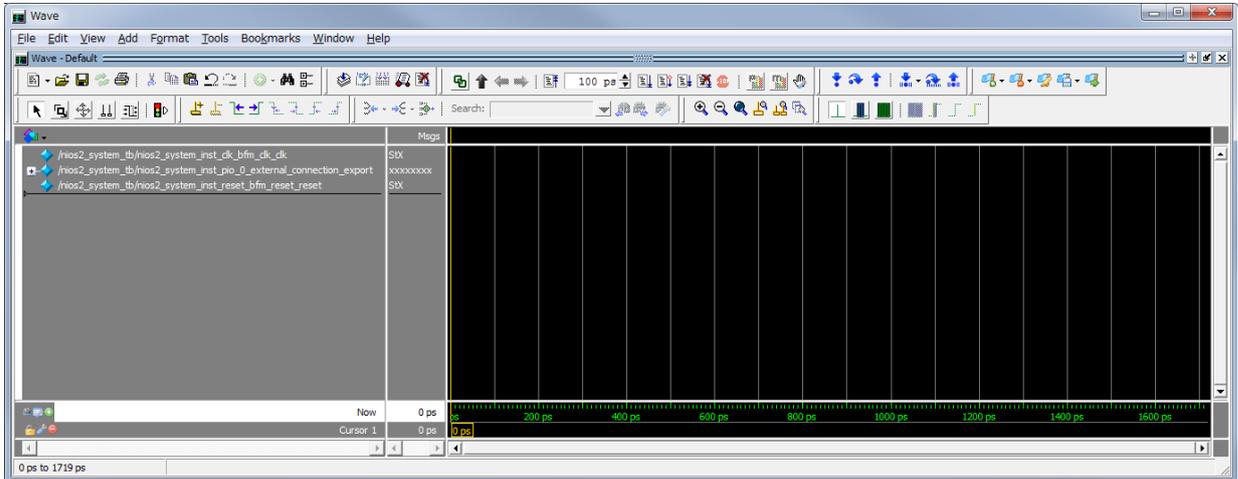
msim\_setup.tcl ファイルは、Qsys を Generate した際に作成されたファイルです。ld コマンドは、msim\_setup.tcl 内で定義されているローカル・コマンドでデバイス・ライブラリ・ファイルのコンパイル、デザイン・ファイルのコンパイル、vsim コマンドの実行を一括で行います。詳細は、msim\_setup.tcl で確認できます。

### 2-3. シミュレーションの実行

- ① 波形表示のための信号を追加します。ここでは、テストベンチ上で定義されているすべての信号を追加します。以下のコマンドを ModelSim® のコンソールから実行してください。

```
VSIM 3>add wave *
```

nios2\_system\_tb.v 内で wire 宣言されている 3 つの信号が追加されます。

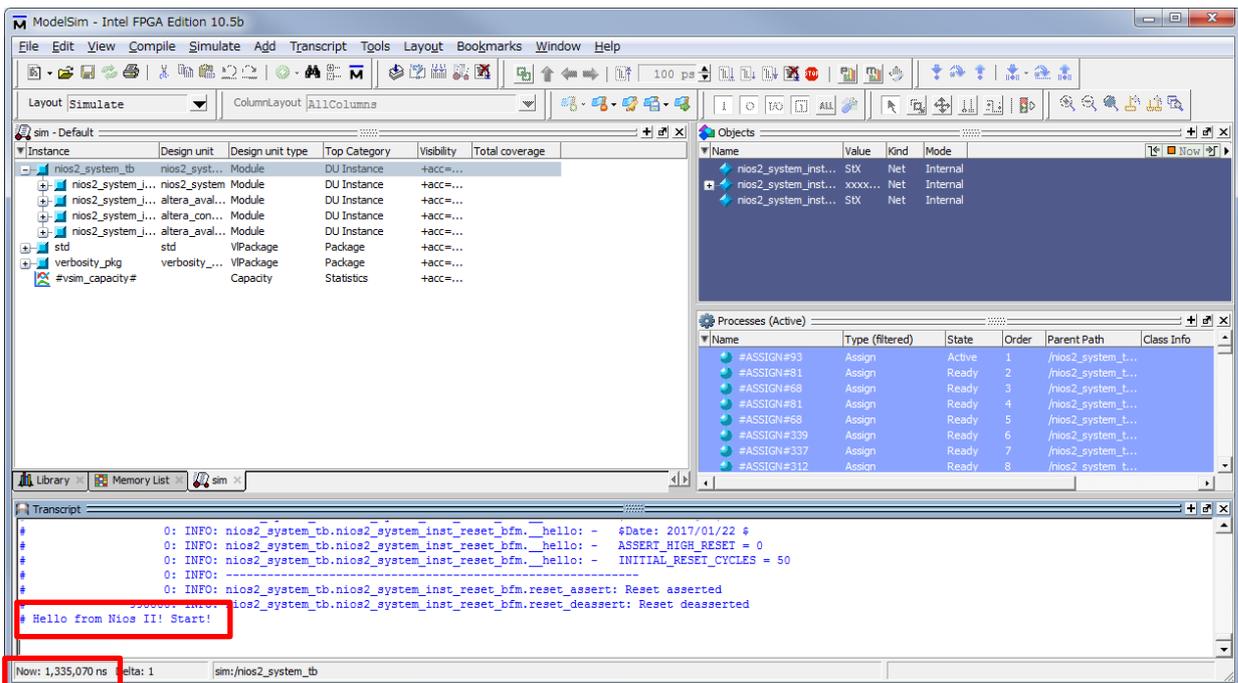


- ② シミュレーションを実行します。ここでは、2ms 時間のシミュレーションを実行します。以下のコマンドを ModelSim® のコンソールから実行してください。

```
VSIM 4>run 2ms
```

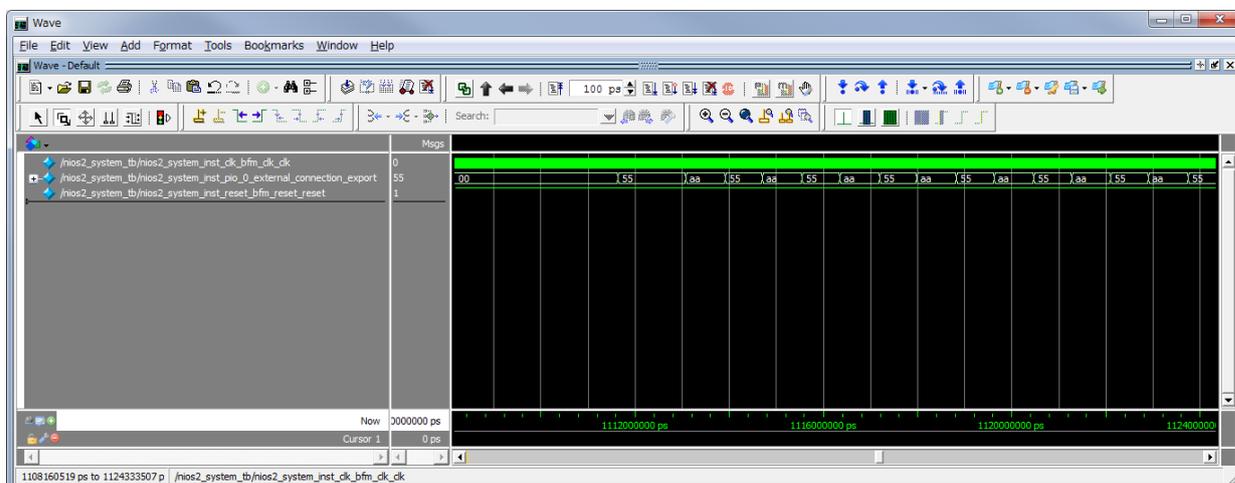
環境にも依存しますが、シミュレーションが終わるまで数分掛かります。実行が 1ms を過ぎたあたりで、printf() 関数の出力「Hello from Nios II! Start!」がコンソールに表示されます。

また、シミュレーションの経過時間を確認することができます。(下のキャプチャでは、約 1.3ms 経過。)



- ③ シミュレーションが完了したら、波形表示を確認します。下図のように波形を拡大し、表示を Hexadecimal にすると、pio\_0\_external\_connection\_export の表示が 0x55 と 0xAA を繰り返していることが確認できます。

これは、別資料「Nios® II はじめてガイド - Nios® II 簡易チュートリアル」で作成したソフトウェアのコードに記述したとおり、0x55 と 0xAA を繰り返していることが確認できました。



## 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 <http://www.altima.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、各メーカー発行の英語版の資料もあわせてご利用ください。