

Nios II ModelSim シミュレーション の実行方法

ver.14

Nios II ModelSim シミュレーションの実行方法

目次

1. はじめに.....	3
2. 操作方法概要	3
2-1. Qsys の設定	3
2-2. Nios II SBT の設定	5
2-3. シミュレーションの実行	7
改版履歴	9

1. はじめに

この資料は、別資料である『Nios II 簡易チュートリアル』を終えた後、Nios[®] II システム・デザインの RTL シミュレーションを ModelSim を使用して行う際の操作マニュアルです。

この資料では、Quartus[®] II 14.0、Nios II 14.0 Software Build Tools for Eclipse（以降、SBT と略）、ModelSim-Altera 10.1e を使用して説明しています。ここでは、ModelSim-Altera を使用していますが、ModelSim-PE、ModelSim-SE でも同様にシミュレーションが可能です。

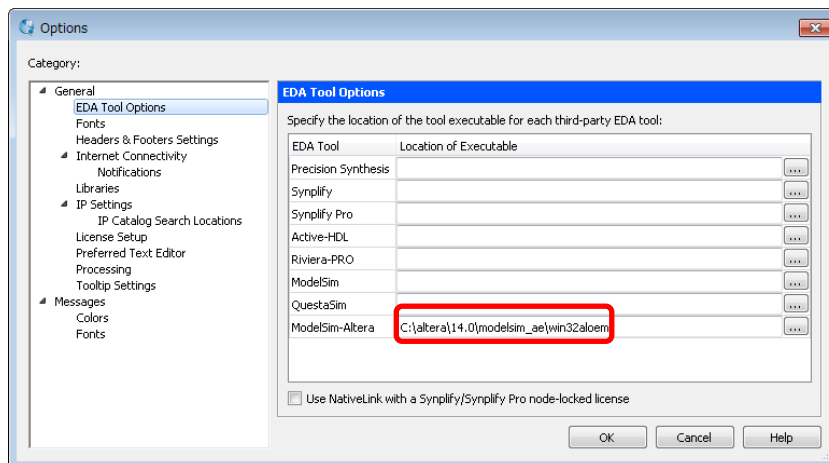
2. 操作方法概要

以下に、Nios II のシステムを ModelSim-Altera にてシミュレーションを行う手順を示します。

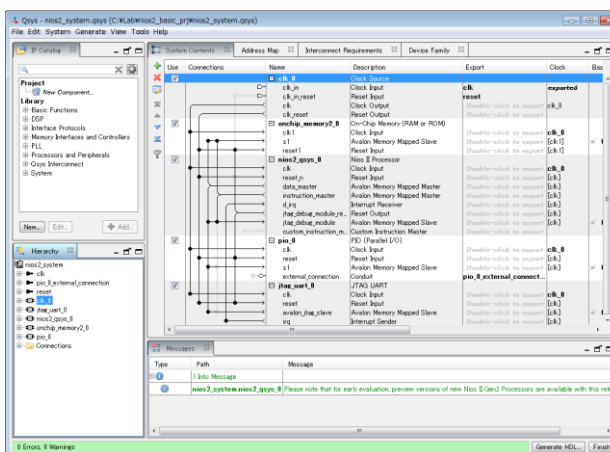
- ① 『Nios II 簡易チュートリアル』を終えた後のハードウェア・デザイン上で、シミュレーションに必要なテストベンチ関連ファイルを Qsys にて生成します。
- ② 『Nios II 簡易チュートリアル』で作成した Nios II SBT のソフトウェア・プロジェクトを開き、シミュレーション用の最適化オプションを設定し、ビルドを実行します。
- ③ Nios II SBT からシミュレーションを実行します。
- ④ 起動した ModelSim-Altera にて、波形表示のためのコマンドを実行します。
- ⑤ シミュレーション実行のコマンドを入力し、シミュレーションを実行します。
- ⑥ シミュレーション結果を確認します。

2-1. Qsys の設定

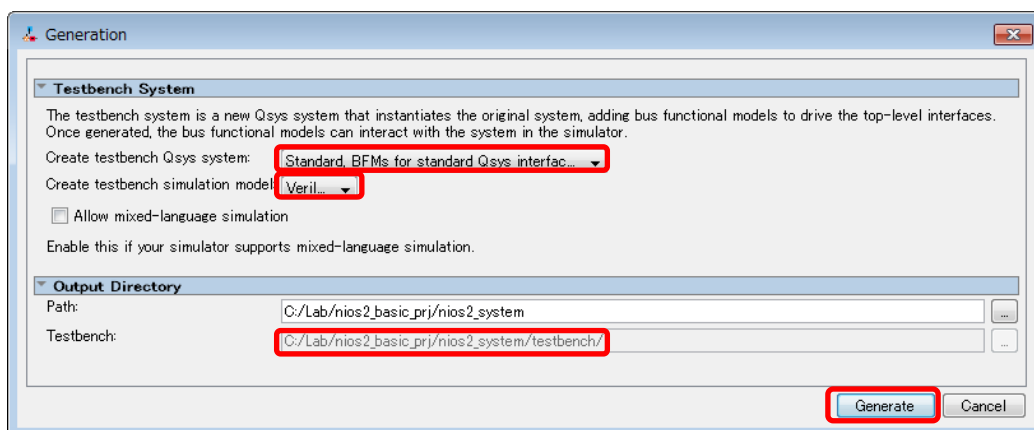
- ① 『Nios II 簡易チュートリアル』では、C:\¥Lab¥nios2_basic_prj フォルダに nios2_basic_lab というプロジェクトを用意しました。この資料でも、同じフォルダ構成を例として進めていきます。
- ② Quartus II を起動し、File メニュー ⇒ Open Project... から nios2_basic_prj フォルダ内の nios2_basic_lab.qpf ファイルを選択し、プロジェクトを開きます。
- ③ 使用する ModelSim-Altera のパスを確認します。Tools メニュー ⇒ Options... を実行し、EDA Tool Options カテゴリを確認します。該当する ModelSim-Altera のパスが設定されていることを確認します。



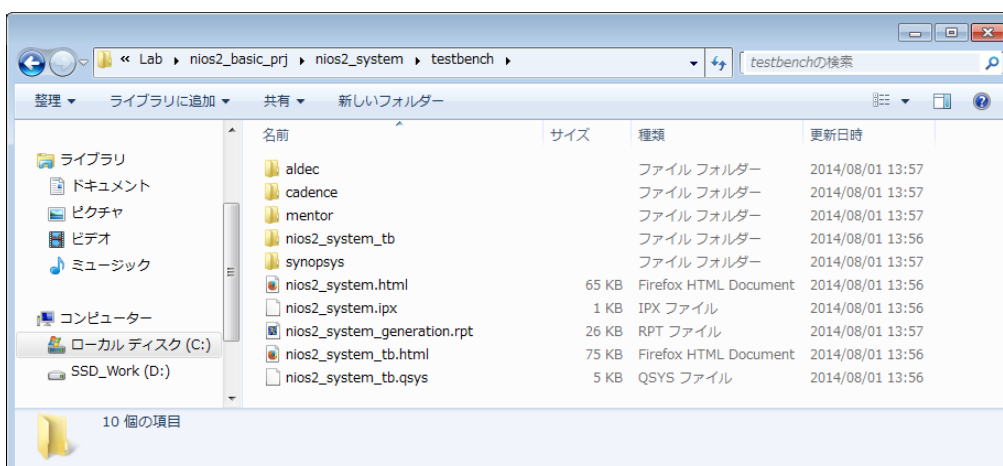
- ④ nios2_basic_lab のプロジェクトで Qsys を起動します。Tools メニュー ⇒ Qsys をクリックし、nios2_system.qsys を選択します。



- 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 14.0 200 at 2014.08.01.13:56:45
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 nios2_system_inst_reset_bfm_reset_reset; // nios2_system_inst_reset_bfm:reset -> nios2_system_inst:reset_reset_n
    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

    nios2_system nios2_system_inst (
        .clk_clk (nios2_system_inst_clk_bfm_clk_clk), // clk_clk
        .reset_reset_n (nios2_system_inst_reset_bfm_reset_reset), // reset_reset_n
        .pio_0_external_connection_export (nios2_system_inst_pio_0_external_connection_export) // pio_0_external_connection.export
    );

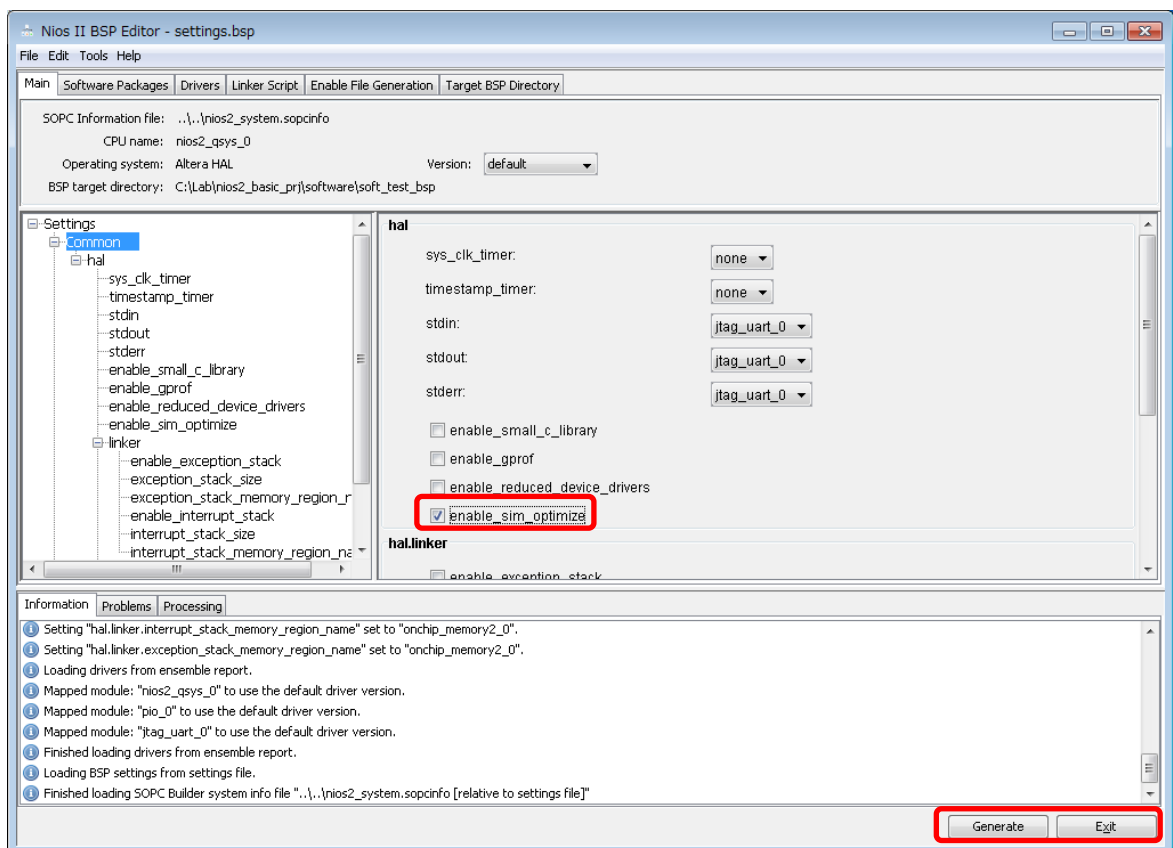
    altera_avalon_clock_source #(
        .CLOCK_RATE (50000000),
        .CLOCK_UNIT (1)
    ) nios2_system_inst_clk_bfm (
        .clk (nios2_system_inst_clk_bfm_clk_clk) // clk_clk
    );

    altera_avalon_reset_source #(
        .ASSERT_HIGH_RESET (0),
        .INITIAL_RESET_CYCLES (50)
    ) nios2_system_inst_reset_bfm (
        .reset (nios2_system_inst_reset_bfm_reset_reset), // reset_reset_n
        .clk (nios2_system_inst_clk_bfm_clk_clk) // clk_clk
    );

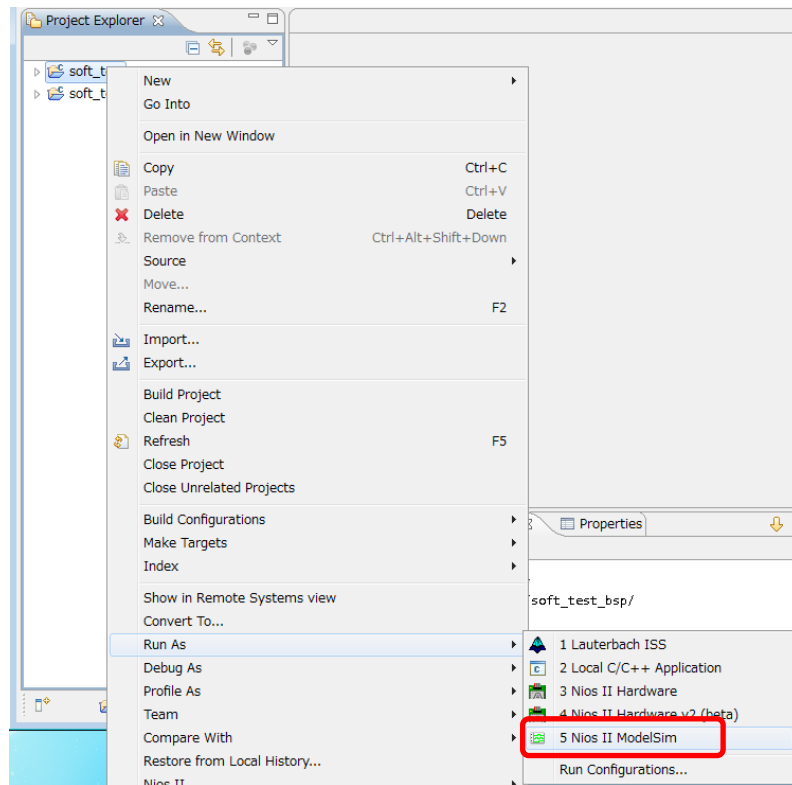
    altera_conduit_bfm_nios2_system_inst_pio_0_external_connection_bfm (
        .sig_export (nios2_system_inst_pio_0_external_connection_export) // conduit.export
    );
endmodule
```

2-2. Nios II SBT の設定

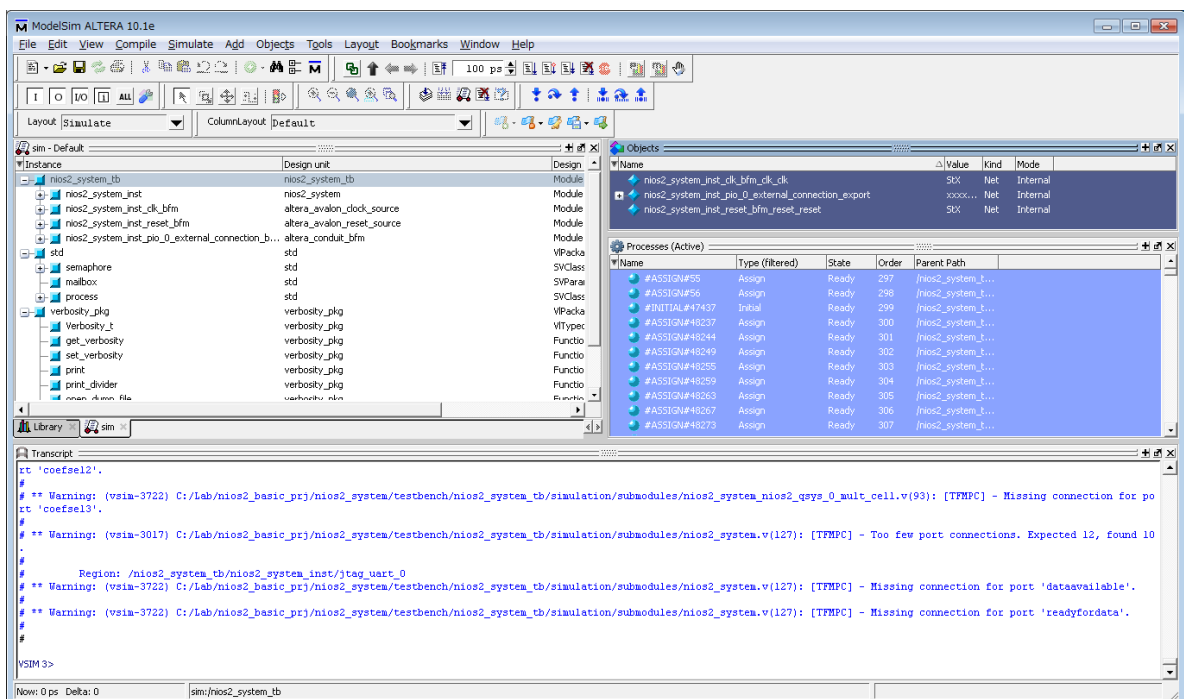
- ① 『Nios II 簡易チュートリアル』を行った際に作成した、Nios II SBT のソフトウェア・プロジェクト soft_test を開きます。
- ② BSP Editor を起動し、シミュレーション用の最適化設定を行います。soft_test_bsp を右クリック ⇒ Nios II ⇒ BSP Editor を実行します。下図のように、enable_sim_optimization にチェックを入れます。この設定を行うことでシミュレーション時間を短縮します。Generate ⇒ Exit で設定を反映させます。



- 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、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-Altera が以下のフォルダで実行されています。

C:\¥Lab¥nios2_basic_prj¥software¥soft_test¥obj¥default¥runtime¥sim¥mentor

コンソール画面で ↑ を押すと、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
```

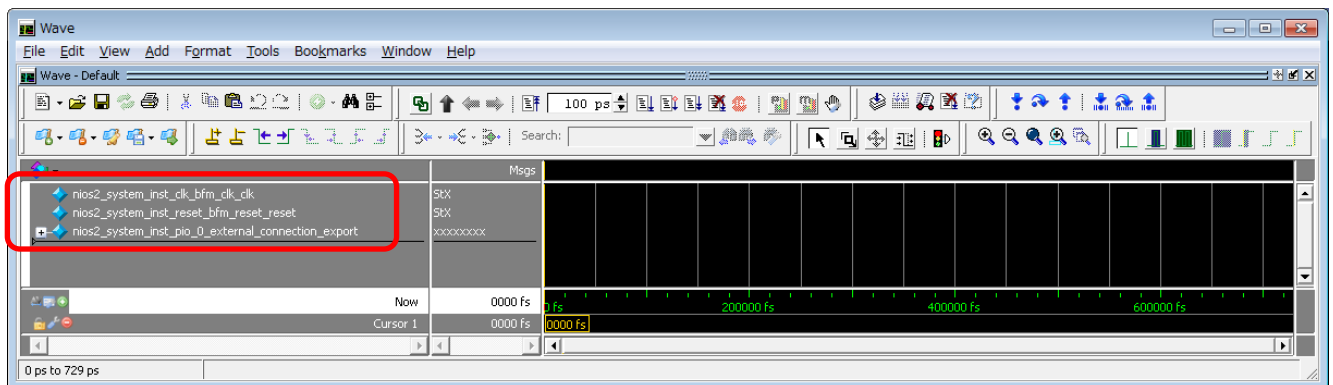
msim_setup.tcl ファイルは、Qsys を Generate した際に作成されたファイルです。ld コマンドは、msim_setup.tcl 内で定義されているローカル・コマンドでデバイス・ライブラリ・ファイルのコンパイル、デザイン・ファイルのコンパイル、vsim コマンドの実行を一括で行います。詳細は、msim_setup.tcl で確認できます。

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 *
```

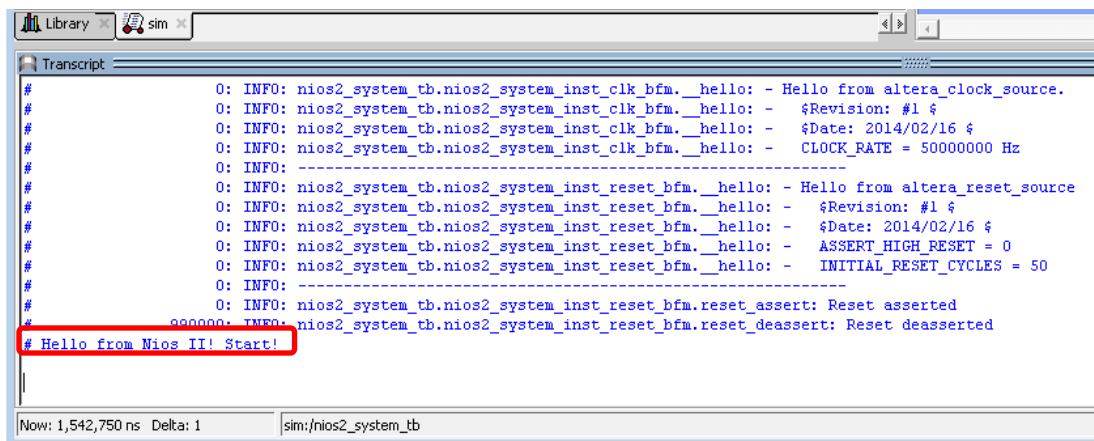
nios2_system_tb.v 内で wire 宣言されている 3 つの信号が追加されます。



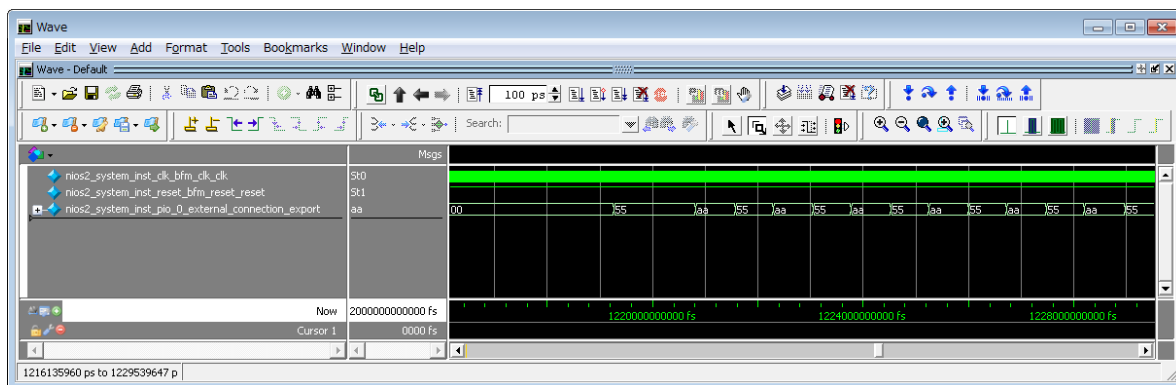
- ② シミュレーションを実行します。ここでは、2ms 時間のシミュレーションを実行します。以下のコマンドを ModelSim のコンソールから実行してください。

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

実行には、2分~3分程度かかります。実行が 1ms を過ぎたあたりで、printf() 関数の出力がコンソールに表示されます。



- 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、0xAA を繰り返していることが確認できます。



改版履歴

Revision	年月	概要
1	2014 年 8 月	新規作成

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。
 株式会社アルティマ ホームページ: <http://www.altima.co.jp> 技術情報サイト EDISON: <https://www.altima.jp/members/index.cfm>
 株式会社エルセナ ホームページ: <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。