# PlanAhead ソフトウェア チュートリアル

## ChipScope を使用したデバッグ

UG677 (v 13.1) 2011 年 3 月 1 日

XILINX®

本資料は英語版 (v 13.1) を翻訳したもので、内容に相違が生じる場合には原文を優先します。
資料によっては英語版の更新に対応していないものがあります。
日本語版は参考用としてご使用の上、最新情報につきましては、必ず最新英語版をご参照ください。

# 改訂履歴

次の表に、この文書の改訂履歴を示します。

| 日付 | 改訂内容 |
|---|---|
| 2011 年 3 月 1 日 | PlanAhead™ ソフトウェアの新規デバッグ フローを説明するために更新 |

# 目次

# PlanAhead ソフトウェア チュートリアル：ChipScope を使用したデバッグ

## 要件

Xilinx® ISE® デザイン ツール フローの基本的な知識

## 目標

このチュートリアルでは、次について説明します。

- PlanAhead™ の ChipScope™ Pro Analyzer 機能を使用して、デザインを素早く簡単にデバッグする方法について説明します。
- PlanAhead および ChipScope ツールで FPGA ロジック デザインでよく発生する問題をデバッグする方法を具体的に説明します。

このチュートリアルを終了すると、次ができるようになります。

- PlanAhead および ChipScope ツールで ILA (Integrated Logic Analyzer) コアを使用し、ChipScope Pro Analyzer を使用してデザインを検証およびデバッグします。
- RTL プロジェクトを作成し、ILA コアを使用してデザインをプローブし、PlanAhead でデザインをインプリメントする方法について理解します。
- PlanAhead ソフトウェアデザイン環境で IP コアのネットリストを生成してカスタマイズします。
- ChipScope Pro Analyzer を使用してデザインをデバッグし、PlanAhead デザイン環境と Spartan®-6 FPGA SP601 評価キット ベース ボード (SP601 プラットフォーム) を使用してデザインを繰り返し実行します。

## はじめに

このチュートリアルを開始する前に、手順を実行するために必要なハードウェアおよびソフトウェア コンポーネントが存在しているかどうか確認し、それらについて理解している必要があります。詳細は、次に説明します。

### 設定要件

このチュートリアルの手順を実行するには、次のソフトウェアおよびハードウェアが必要です。

- Xilinx ISE® Design Suite 13.1 (Logic、DSP、Embedded、または System Edition)

- Spartan-6 FPGA SP601 評価キット ベース ボード SP601 ボードに関する情報については、「その他のリソース」のリンクを参照してください。
  このチュートリアルでは、SP605 または ML605 ボードを使用することもできます。詳細は、7 ページの「ボード サポートおよびピン配置情報」を参照してください。
- SP601 評価キットには USB ケーブルおよび電源ケーブルが含まれます。

次の図 1 は、SP601 ボードです。

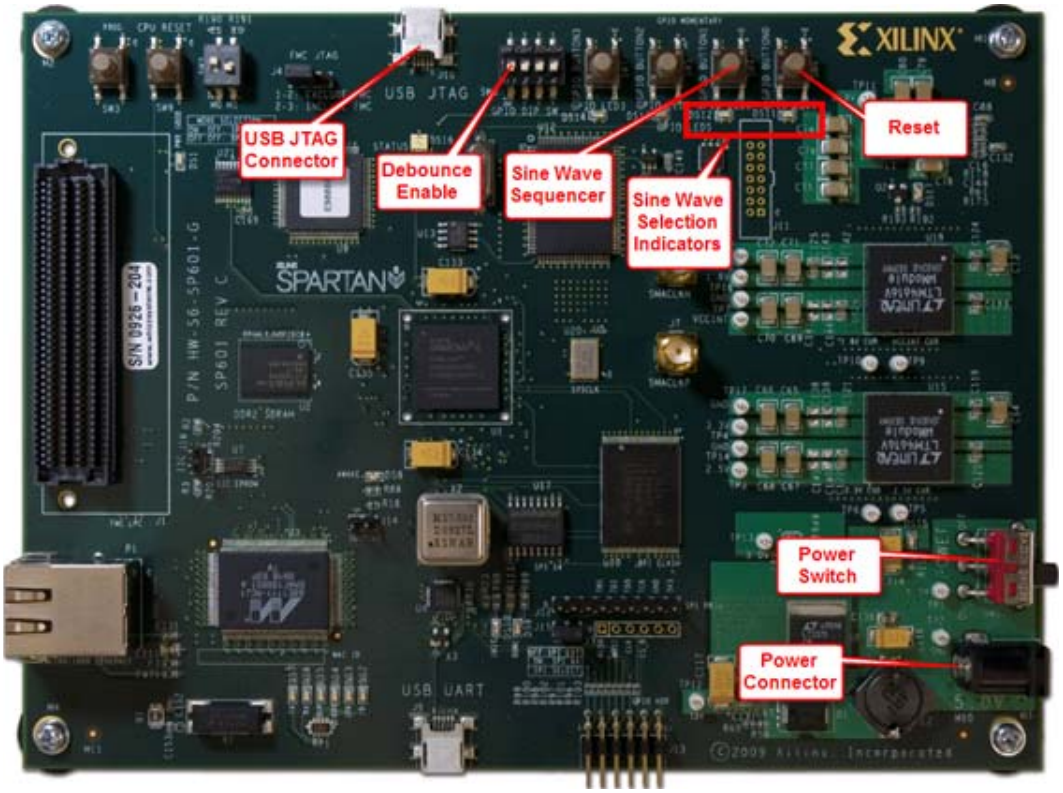


図 1 : SP601 ボード

## チュートリアルのデザイン コンポーネント

チュートリアル デザインには、次が含まれます。

- シンプルな制御ステート マシン
- 複数のサイン波ジェネレーター
- プッシュ ボタン (GPIO_BUTTON)
- DIP スイッチ (GPIO_SWITCH)
- LED ディスプレイ (GPIO_LED)
- プッシュ ボタン スイッチ :デバウンスおよび制御ステート マシン回路への入力になります。ボタンを押すと High から Low への遷移パルスが生成されます。生成された出力パルスは、それぞれステート マシンへの入力として使用されます。
- DIP スイッチ :デバウンス回路をイネーブルまたはディスエーブルにします。

- デバウンス 回路 :この例の場合、イネーブルで High から Low へのパルスまたは遷移が提供されます。ボタンを押してから放すと、一連のスパイクやグリッチが削除されます。
- サイン波シーケンサー ステート マシン :2 つのプッシュ ボタン スイッチからの入力パルスをキャプチャしてデコードします。00、01、10、11 (0 ～ 3) 間のサイン波選択回路およびインジケーター回路を提供します。
- LED ディスプレイ :GPIO_LED_0 および GPIO_LED_0 はステート マシンの出力からの選択ステータスを表示し、それぞれ高、中、低周波の異なるサイン波周波数を表します。
- チュートリアル デザイン ファイル : デザイン ファイルの読み込み方は、8 ページの「はじめに」を参照してください。

## ボード サポートおよびピン配置情報

**メモ** : このチュートリアルでは、次の 2 つのザイリンクス ボードもサポートされています。SP605 および ML605このチュートリアルの対象を SP605 か ML605 に変更する場合は、表 1 のピン配置情報を参照してください。

表 1 : SP605 および ML605 ボードのピン配置情報

| | ピン配置の位置 | | | ファンクション |
|---|---|---|---|---|
| | **SP601** | **SP605** | **ML605** | |
| **CLK_N** | K16 | K22 | H9 | クロック |
| **CLK_P** | K15 | K21 | J9 | クロック |
| **GPIO_BUTTONS[0]** | P4 | F3 | A19 | リセット |
| **GPIO_BUTTONS[1]** | F6 | G6 | G26 | サイン波シーケンサー |
| **GPIO_SWITCH** | D14 | C18 | D22 | デバウンス回路セレクター |
| **LEDS_n[0]** | E13 | D17 | AC22 | サイン波選択[0] |
| **LEDS_n[1]** | C14 | AB4 | AC24 | サイン波選択[1] |
| **LEDS_n[2]** | C4 | D21 | AE22 | 予約済み |
| **LEDS_n[3]** | A4 | W15 | AE23 | 予約済み |

# 手順 1 : PlanAhead ソフトウェアでの RTL プロジェクトの作成および インプリメンテーション

この手順では、次を実行します。

- チュートリアルのソース ファイルを解凍し、PlanAhead ソフトウェアで開きます。
- New Project ウィザードで新規プロジェクトを作成します。
- デザインを合成します。

## はじめに

1. C:\ ドライブに **ChipScope_PlanAhead** というフォルダを作成します。
2. チュートリアル デザインのソース ファイルを見つけます。

   このチュートリアルでは、PlanAhead ソフトウェアに含まれるプロジェクト例かダウンロード可能な圧縮 ZIP ファイルに含まれるサンプル デザイン データを使用します。

   デザイン データのロケーション :

   - http://japan.xilinx.com/support/documentation/dt_planahead_planahead13-1_tutorials.htm
   - *<ISE_install_area>*/PlanAhead/testcases/PlanAhead_Tutorial.zip

   チュートリアルおよびデザイン ファイルは、ソフトウェア リリースごとにアップデートされることがあり、最新バージョンはウェブサイトから入手できるようになっています。
3. チュートリアルのソース ファイルを **ChipScope_PlanAhead** で解凍します。
4. 解凍したら、ChipScope_PlanAhead/PlanAhead_Tutorial/Sources/chipscope/pa_cs_tutorial/src は図 2 のようになります。

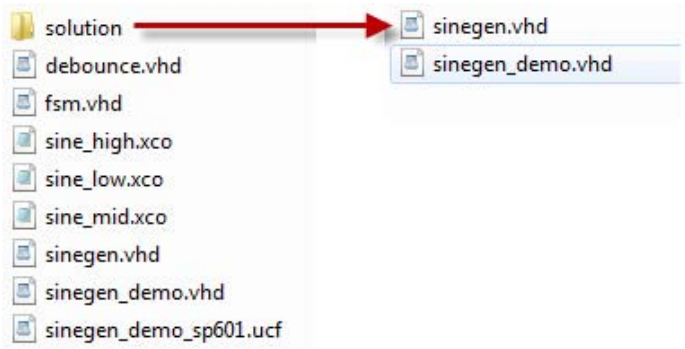


図 2 : チュートリアル デザインのファイル セット

## PlanAhead ソフトウェアの New Project ウィザードを使用したプロジェクトの作成

プロジェクトを作成するには、New Project ウィザードでプロジェクト名を指定し、RTL ソース ファイルおよび制約を追加し、ターゲット デバイスを指定します。

1. PlanAhead ソフトウェアを起動します。

2. Getting Started ページで [Create New Project] のリンクをクリックし、New Project ウィザードを開始します。
3. [Project Name] ページで新規プロジェクト名に pa_step1 を指定し、ディレクトリを C:\ChipScope_PlanAhead に指定します。
4. [Design Source] ページで [Specify RTL Sources] をオンにします。
5. [Add Sources] ページで次を指定します。
   a. [Add Files] ボタンをクリックします。
   b. [Add Source Files] ページでプロジェクト デザイン ファイルを保存した **src** ディレクトリを選択します。
   c. すべての VHDL ソース ファイルを選択し、[OK] をクリックします。
   d. ファイルが追加されたことを確認したら、[Next] をクリックします。
6. [Add Existing IP] ページで次を指定します。
   a. [Add Files] ボタンをクリックします。
   b. [Add Configurable IP] ページで **src** ディレクトリを選択します。
   c. すべての XCO ソース ファイルを選択し、[OK] をクリックします。
   d. ファイルが追加されたことを確認したら、[Next] をクリックします。
7. [Add Constraints (optional)] ページに UCF ファイルが自動的に表示されます。デフォルトのまま、[Next] をクリックします。
8. [Default Part] ページで SP601 ボードの **xc6slx16csg324-2** を指定します。パーツ リストのすぐ上の検索ツールを使用すると、簡単に見つけることができます。
9. [New Project Summary] ページを確認します。データが指定どおりに表示されていることを確認します。

**メモ**：プロジェクトの初期化には 1 ～ 2 分かかることがあります。

New Project ウィザードを終了したら、PlanAhead ソフトウェアのメイン ウィンドウから Project Manager を使用し、IP を追加し、デザインを合成します。

## デザインの合成

1. 左のパネルで [Synthesize] ボタンをクリックします。
2. [Select Top Module] ページで **sinegen_demo** を選択します。

   **メモ**：合成を実行中は、進捗状況を示すインジケーターが表示されます。これには、数分かかります。
3. [Synthesis Completed] の画面が表示されたら、[Cancel] をクリックします。デザインは後でインプリメントします。
4. [File] → [Save Project As] をクリックし、プロジェクト名を `pa_step2` にして保存します (プロジェクトに別の名前を付けて保存しておくと、すべての手順を一気に終了しない場合などに便利です)。

# 手順 2 : ChipScope ツールを使用した PlanAhead ソフトウェア デザインのデバッグ

ChipScope Analyzer ILA コアをデザインに追加するには、PlanAhead と ChipScope ツールを連動させます。

フローがシンプルになったので、元の RTL ソース コードを修正しなくてもデザインをプローブできます。ここでは、次を実行します。

- デバッグ ネットのプロジェクトへの追加
- Set Up ChipScope ウィザードの実行
- デザインをインプリメントして開く
- ビットストリームの生成

## デバッグ ネットのプロジェクトへの追加

`pa_step2` プロジェクトで次を実行します。

1. [Netlist Design] ドロップダウン リストから [Open Netlist Design] をクリックします。
2. [Open Netlist Design] はデフォルトのままにします。
3. [Netlist] タブをクリックします。
4. デバッグする次のネットを選択します (図 3)。
   - GPIO_BUTTONS_db(2)
   - GPIO_BUTTONS_dly(2)
   - GPIO_BUTTONS_re(2)
   - sine(20)
   - sineSel(2)
   - GPIO_BUTTONS_0_IBUF
   - GPIO_BUTTONS_1_IBUF

   **メモ**：これらの信号は、このデザインの主な動作を示しており、次の手順でデザインを検証およびデバッグするために使用されます。
5. 選択したネットを右クリックし、[Add to ChipScope Unassigned nets] をクリックします。

**メモ**：[ChipScope] タブでは、選択したばかりの割り当てのないネットが表示されます。

図 3 は、上記の手順を示しています。

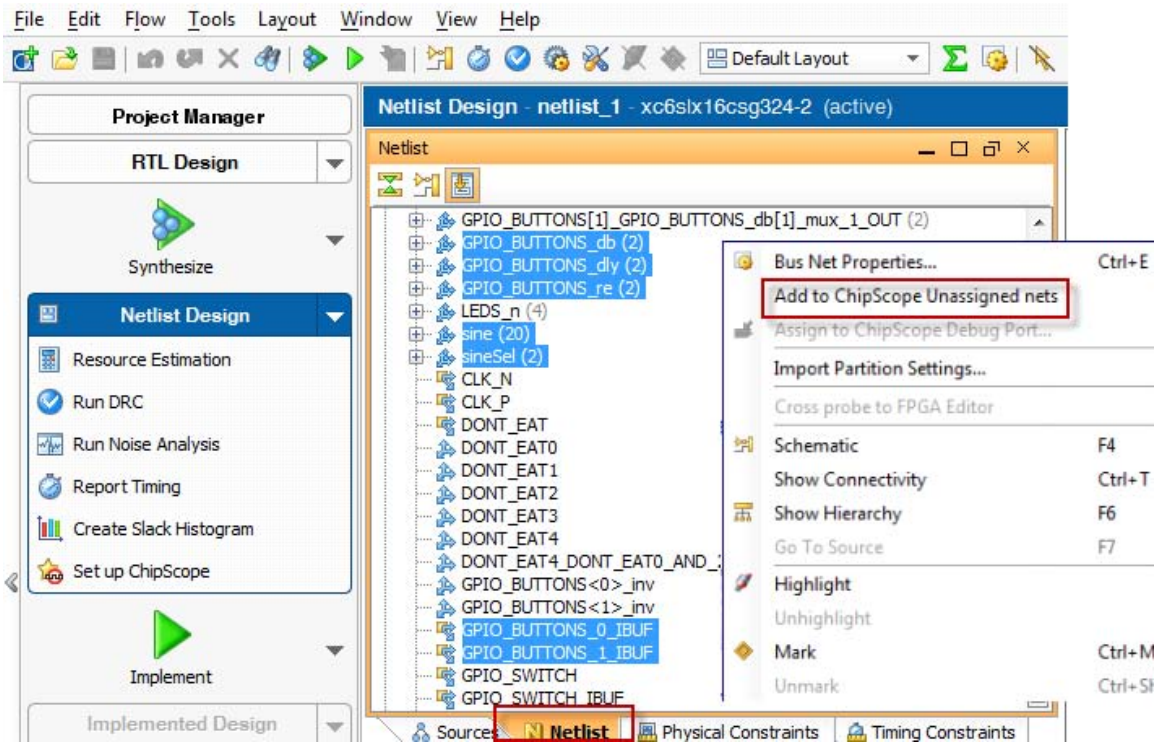


図 3 : [Netlist] タブからのネットの追加

## Set Up ChipScope ウィザードの実行

1. [Netlist Design] ドロップダウン リストから [Set Up ChipScope] をクリックします。
2. Set Up ChipScope ウィザードが開きます。ウィザードをクリックしていき、ChipScope Analyzer デバッグ コアを作成します。デフォルトの設定を使用してください。

## デザインをインプリメントして開く

1. PlanAhead メイン ウィンドウの左のパネルの [Implement] ボタンをクリックします。
2. [Save Project] ポップアップ メニューで [Save] を選択します。
3. インプリメンテーション プロセスが終了すると、[Implementation Completed] ダイアログ ボックスが表示されます。[Open Implemented Design] をオンにし、[OK] をクリックします。

   **メモ**：インプリメンテーションには、数分かかります。
4. インプリメントされたデザインを開く前にネットリスト　デザインを閉じるかどうかを尋ねるメッセージが表示されます。[Yes] をクリックします。

図 4 は上記の手順を示しています。

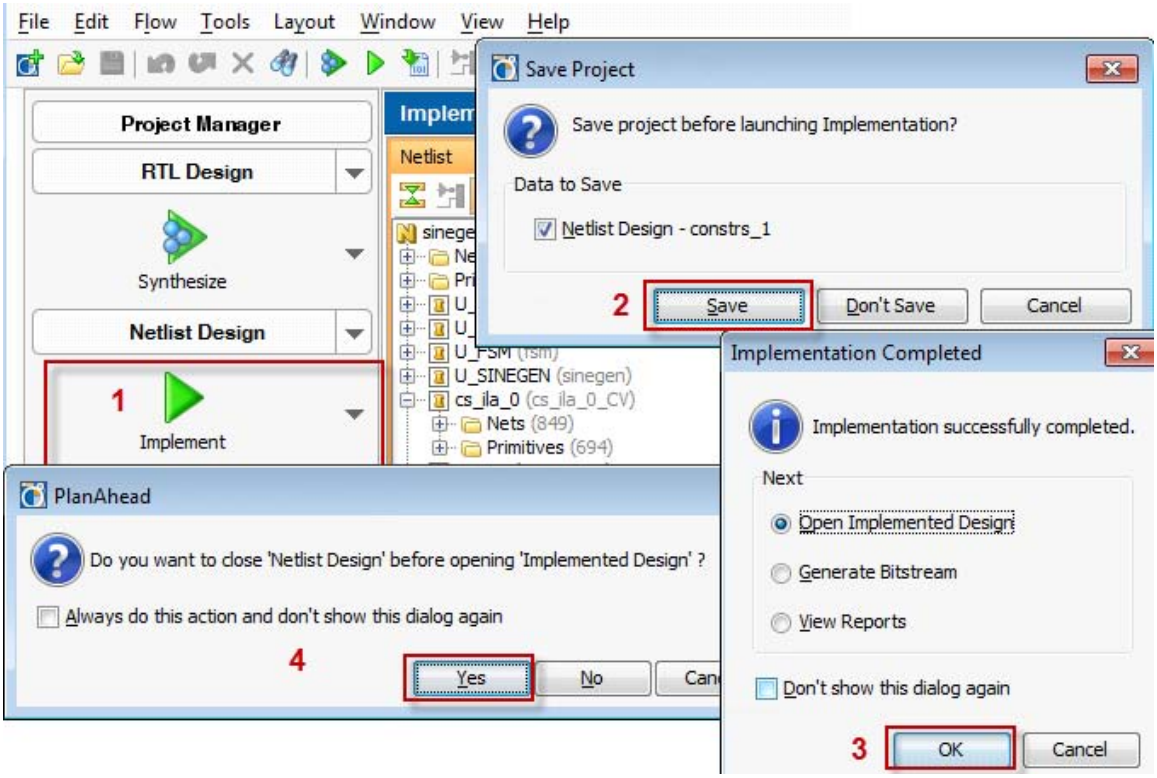


図 4 : デザインをインプリメントして開く手順

### ビットストリームの生成

1. 左のペインで [Program and Debug] の下の [Generate Bitstream] をクリックします。
2. [Generate Bitstream] ダイアログ ボックスが開きます。[OK] をクリックし、ビットストリームの生成を開始します。
3. プロセスが終了すると、[Bitstream Generation Completed] ポップアップが表示されます。

## 手順 3 : ChipScope ツールを使用したハードウェアのデバッグ

この手順を実行すると、次ができるようになります。

- ChipScope を使用したデザインをデバッグします。
- デザインに少し変更を加えて、回路の問題を発見して修正します。
- デザイン データをトリガーおよびキャプチャするために有益な手法について学びます。

### サイン波ジェネレーターの動作確認

設定をしてから、ChipScope Pro Analyzer を使用してサイン波ジェネレーターが正しく動作しているかどうか確認します。確認するのは、次の 2 つです。

- すべてのサイン波の選択が正しいかどうか
- 選択ロジックが正しく動作しているかどうか

## 設定

1. SP601 ボードが正しく設定されていることを確認します。
   - USB ケーブルはボードの USB JTAG コネクターからシステムの USB ポートに接続されているる必要があります (ボードの USB UART コネクターは使用しないでください)。
   - ボードがプラグインされ、電源が投入されています。
   - すべての DIP スイッチの位置が OFF に設定されています。
2. PlanAhead ソフトウェアの [Program and Debug] ドロップダウン リストから [ChipScope Analyzer] をクリックします。
3. ChipScope Analyzer で JTAG チェーンを USB ケーブルに設定し、次の通信パラメータを設定します。
   a. [JTAG Chain] → [Xilinx Platform USB Cable] をクリックします。
   b. プラットフォームの [USB Cable Parameters] ダイアログ ボックスが表示されます。速度が **3MHz**、ポートが **USB21** に設定されていることを確認します。
4. ChipScope Pro Analyzer ウィンドウの一番下のペインに表示される情報から、JTAG チェーンへ接続されているかどうか確認します (図 5)。

   **メモ** : ESN オプション番号は図とは異なります。この番号はボードごとに異なります。

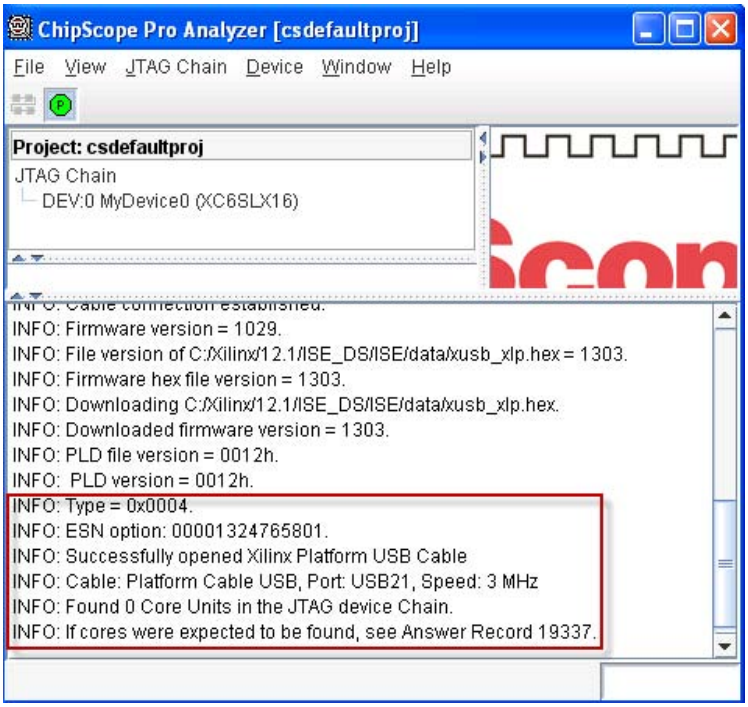


図 5 : JTAG チェーンの接続データ

5. デバイスをコンフィギュレーションします。
   a. ChipScope Pro Analyzer の左上のペインの [Project: csdefaultproj] の下にリストされる JTAG チェーン デバイスを右クリックし、[Configure] をクリックします。
   b. [JTAG Configuration] ダイアログ ボックスが開きます。このダイアログ ボックスでは、あらかじめ作成しておいた BIT ファイルを使用してデバイスをプログラムします。BIT およ

び CDC ファイルのディレクトリは PlanAhead から ChipScope Pro Analyzer に伝達されます。このため、すべての設定をデフォルト値のままにしておきます。

c. これでデバイス コンフィギュレーションと ILA コアを ChipScope Pro Analyzer のメイン ウィンドウで確認できるようになりました (図 6)。

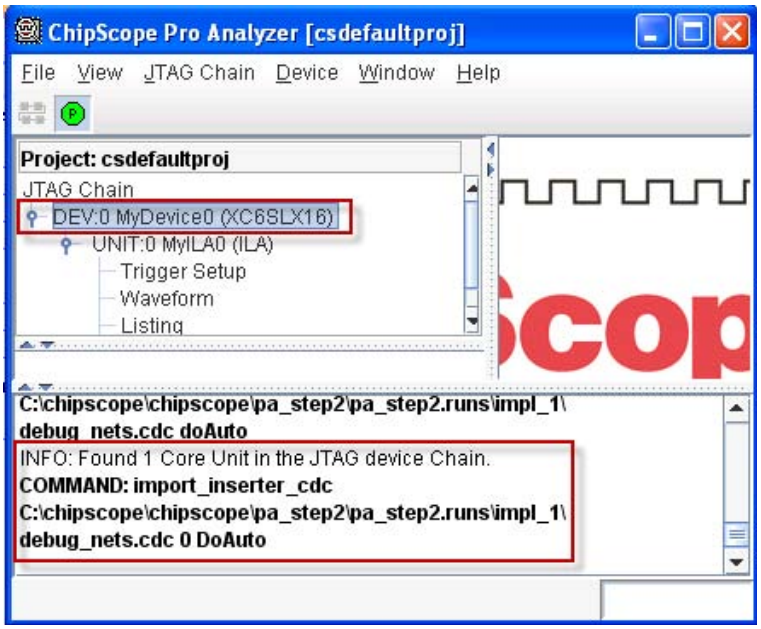


図 6 : デバイス コンフィギュレーションおよび ILA コア情報

## サイン波の動作確認

1. [DEV:0 MyDevice0 (XC6SLX16)] を展開し (図 6)、[Trigger Setup] をダブルクリックします。右上のペインに [Trigger Setup] が表示されます。
2. [Waveform] をダブルクリックし、波形表示を右上のペインに追加します。
3. ツールバー メニューで [T!] をクリックし、トリガーしてデータをキャプチャします。
4. [Waveform] ウィンドウで **sine** 信号の動作を確認します。

## サイン波の表示

1. [DEV:0 MyDevice0 (XC6SLX16)] を展開し、[Bus Plot] をダブルクリックして [Bus Plot] ウィンドウを開きます。
2. [Bus Plot] ウィンドウで [/sine] チェック ボックスをオンにしてサイン波を表示します。

   波形はサイン波のようには見えません。これは、次の方法で基数設定を 16 進数から符号付き 10 進数に変更すると正しく表示されます。

## サイン波の表示方法の変更

基数設定を変更して表示を正しくするには、[Trigger Immediate] 機能 (T! ボタン) を使用し、高、中、低周波数のサイン波のバス プロットを確認し、それぞれの設定を変更します。

1. 画面左側の [Signals] ウィンドウで [/sine] を右クリックし、[Bus Radix] → [Signed Decimal] をクリックします。

2. サイン波選択ステート マシンが正しく動作しているかどうかを確認しているには、SP601 ボードで次の表 2 の手順に従ってください。表のボード コンポーネントの識別には、6 ページの図 1 を参照してください。

   **メモ**：サイン波選択の際に、LED が予測どおりに点灯しないことがあります。これについては、次のセクションでデバッグします。ここからは、LED 選択ごとに正しいサイン波が表示されているかどうかを確認します。

表 2 : サイン波選択のシーケンス

| SP601 ボードの設定テスト | ChipScope Pro Analyzer でのトリガーとデータ キャプチャー | テスト結果の確認 |
|---|---|---|
| 1. サイン波形選択のインジケーター (LED) が両方ともオフ (0,0) であることを確認します。オフでない場合は、オフになるまで [Sine Wave Sequencer] ボタンを押します。 | [T!] (Trigger Immediately) ボタンをクリックし、高周波のサイン波のバス プロットを確認します。 | [Bus Plot] ウィンドウに高周波のサイン波が表示されていることを確認します。 |
| 2. サイン波選択インジケーター (LED) 2 つがオフとオン (0,1) になるまで、ボードの [Sine Wave Sequencer] ボタンを押します。 | [T!] をクリックし、中周波数のサイン波のバス プロットを確認します。 | [Bus Plot] ウィンドウに中周波数のサイン波が表示されていることを確認します。 |
| 3. サイン波選択インジケーター (LED) 2 つがオンとオフ (1,0) になるまで、ボードの [Sine Wave Sequencer] ボタンを押します。 | [T!] をクリックし、低周波数のサイン波のバス プロットを確認します。 | [Bus Plot] ウィンドウに低周波数のサイン波が表示されていることを確認します。 |
| 4. サイン波選択インジケーター (LED) 2 つがオンとオン (1,1) になるまで、ボードの [Sine Wave Sequencer] ボタンを押します。 | [T!] をクリックし、混合サイン波のバス プロットを確認します。 | [Bus Plot] ウィンドウに混合されたサイン波が表示されていることを確認します。 |

## サイン波シーケンサー ステート マシンのデバッグ

サイン波の表示を変更したため、[Sine Wave Sequencer] ボタンを押したときに LED が順番どおりに点灯しないことがあります。

ボタンを押すごとに、GPIO_BUTTONS_re[1] 信号に 1 サイクル幅のパルスが送られます。これが複数ある場合は、LED が規則的に点灯しなくなります。このチュートリアルのセクションでは、ChipScope を使用してサイン波シーケンサー ステートマシンをプローブし、グリッチの原因を表示して、それを修正します。

実際のデバッグ プロセスを始める前に、サイン波シーケンサー ステート マシンについて理解している必要があります。

### サイン波シーケンサー ステート マシンの概要

サイン波シーケンサー ステート マシンでは、4 つのサイン波の中から 1 つを選択して、デザインの最上位レベルの `sine` 信号に駆動します。ステート マシンには、入力 1 つと出力 1 つがあります。図 7 は、ステート マシンの回路図エレメントを示しています。次の説明を読んで、ステート マシンのグリッチを表示して修正する際には、この図を参照してください。

- 入力は `button` というスカラー信号です。button 入力が 1 の場合、ステート マシンは 1 つのステートから次のステートへ進みます。

- 出力は Y という 2 ビット信号のベクターで、4 つのサイン波ジェネレータのどれが選択されているかを示します。

入力信号の button は、最上レベルの信号 GPIO_BUTTONS_re[1] に接続されます。これは、[Sine Wave Sequencer] ボタンの High から Low への遷移インジケーターです (6 ページの図 1)。出力信号の Y は、サイン波を選択する最上位レベルの信号 sineSel に接続されます。

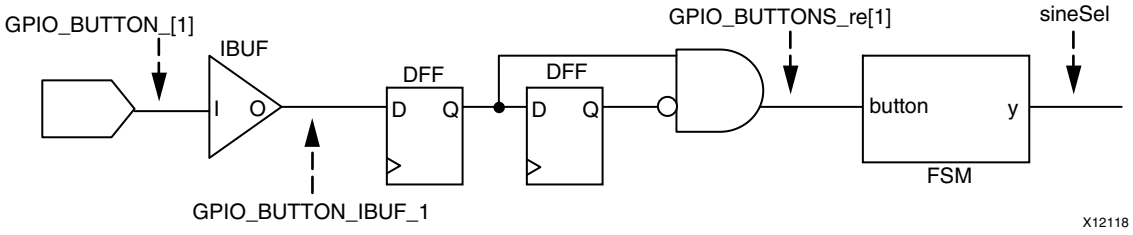


図 7 : サイン波シーケンサーのボタンの回路図

## ステート マシンのグリッチの表示

### ステート マシンへの入力の表示

この手順では、図 7 に示すステート マシンへの入力 (GPIO_BUTTONS_re[1]) を確認し、この信号がグリッチの原因となっているかどうかを検証します。まず、トリガーを次のように設定します。

1. [Trigger Setup] ウィンドウの [Match] エリアで GPIO_BUTTONS_re[1] と GPIO_BUTTONS_re[0] を含む [Match Unit] の行を広げます。
2. [GPIO_BUTTONS_re[1]] 行の [Value] 列を選択し、「R」(立ち上がりエッジ) と入力して Enter を押します。これにより、GPIO_BUTTONS_re[1] でトリガー ポートのマッチ ユニットが 1 サイクル幅のパルスを探すように設定されました。

   メモ : GPIO_BUTTONS_re[1] 信号に値 1 が代入されると、サイン波シーケンサー ステート マシンのステートが次のステートに変わります。
3. [Trigger Setup] ウィンドウの [Trig] エリアで [Trigger Condition Equation] フ列の中をクリックします。表示されるダイアログ ボックスで、正しいマッチ ユニットがイネーブルになっていることを確認します。イネーブルになっている場合は、図 8 のように青色でハイライトされます。

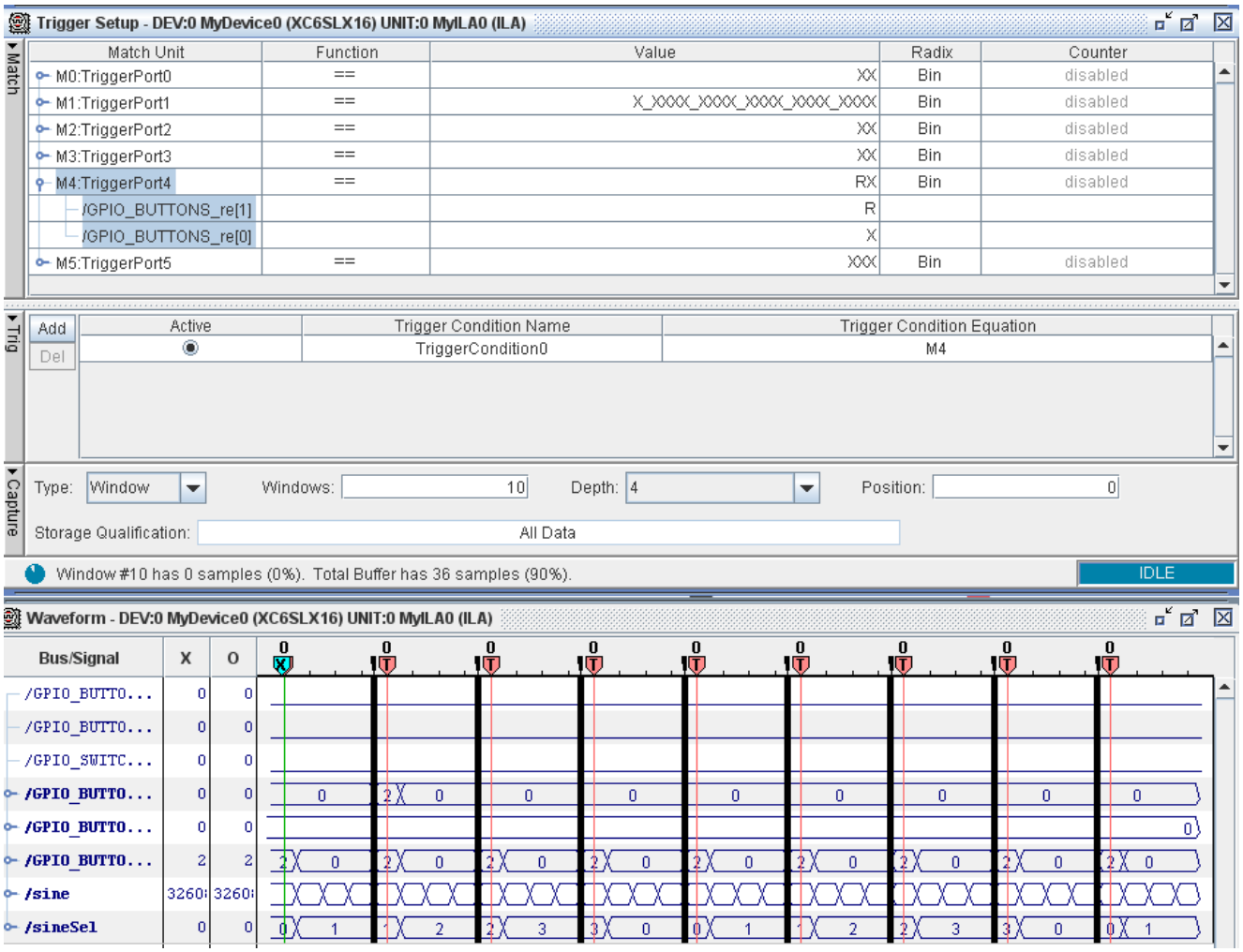


図 8 : 表示設定およびステート マシンのグリッジの例

4. [Trigger Setup] ウィンドウの [Capture] エリアで設定を次のように変更します (図 8)。
   - **Windows** = 10
   - **Depth** = 4

   メモ : [Depth] の設定により、各ウィンドウのサンプル数が制御されます。

### データのキャプチャーと表示

図 9 のツールバー ボタンを参照し、次の手順を実行します。


図 9 : ツールバー ボタン: [Apply Settings and Arm Trigger] および [Stop Acquisition]

1. ツールバーの [Apply Settings and Arm Trigger] ボタンをクリックします。
2. [Trigger Setup] および [Waveform] ウィンドウが表示されていることを確認します。
3. SP601 ボードの [Sine Wave Sequencer] ボタンを押します。キャプチャ設定のすぐ下に表示されるステータス バーから、キャプチャされたウィンドウ数を確認します。ステータス バーは、図 8 に表示されています。ボタンを 1 度押しただけで複数のウィンドウがキャプチャされた場合は、GPIO_BUTTON[1] 入力に問題があります。ウィンドウが 1 つしかキャプチャされなかった場合は、もう 1 度ボタンを押します。このグリッチを再現するには、ボタンを何度か押す必要のあることもあります。

   ボタンを 1 度押して複数のウィンドウが生成されたら、次の手順に進みます。
4. ツールバーの [Stop Acquisition] ボタンをクリックし、キャプチャーされたデータを表示します。

   複数のウィンドウそれぞれの GPIO_BUTTON_re[1] 信号で Low から High の遷移が表示されます。ボタンを 1 度押すと、Low から High への遷移が 1 つだけ表示されます。これは、GPIO_BUTTON[1] 入力信号に何らかの問題があることを意味しています。

### デザインへのボタン入力の表示

デバッグ プローブを GPIO_BUTTON[1] 入力信号自体に接続しても、上記の問題はトラブルシュートできません。GPIO_BUTTON[1] 入力信号は、FPGA ファブリックから直接アクセス可能ではない PAD 信号です。その代わりに、GPIO_BUTTON_IBUF_1 信号 (GPIO_BUTTON[1] 入力信号の入力バッファの出力へ接続される) の Low から High の遷移 (立ち上がりエッジ) でトリガーをする必要があります (16 ページの図 7)。

1. [Trigger Setup] ウィンドウの [Match] エリアで GPIO_BUTTONS_IBUF_1 と GPIO_BUTTONS_IBUF_1 を含む [Match Unit] の行を広げます。
2. [GPIO_BUTTONS_IBUF_1] 行の [Value] 列に R と入力し、Enter を押し、トリガー ポートのマッチ ユニットを設定して、GPIO_BUTTONS_IBUF_1 信号の 1 サイクル幅のパルスを探します。
3. [Trigger Setup] ウィンドウの [Trig] エリアで [Trigger Condition Equation] フ列の中をクリックします。表示されるダイアログ ボックスで、正しいマッチ ユニットがイネーブルになっていることを確認します。

   前述したように、グリッチは GPIO_BUTTONS_IBUF_1 信号の複数の Low から High への遷移として表れ、断続的に発生します。グリッチを検出するにはボタンを何度も押す必要のあることがあるので、ChipScope Pro Analyzer ツールのトリガー実行モードを [Repetitive] に設定しておきます。この設定をしておくと、ボタンを繰り返し押しやすくなり、[Waveform] ウィンドウのイベントを探しやすくなります。
4. ツールバーのドロップダウン メニューで [Trigger Run Mode] を [Repetitive] に設定します。
5. [Trigger Setup] ウィンドウの [Capture] エリアで設定を次のように変更します。
   - Windows = 1
   - Depth = 1024

- Position = 512

6. [Apply Settings and Arm Trigger] ボタンをクリックします。
7. GPIO_BUTTONS_1_IBUF 信号の複数の遷移が表示されるまで、ボードで [Sine Wave Sequencer] ボタンを押します。これは、入力で発生しているグリッチを視覚化する作業です。グリッチの例は、図 10 を参照してください。

   **メモ** : 信号グリッチは図と同じ位置に表示されないこともあります。

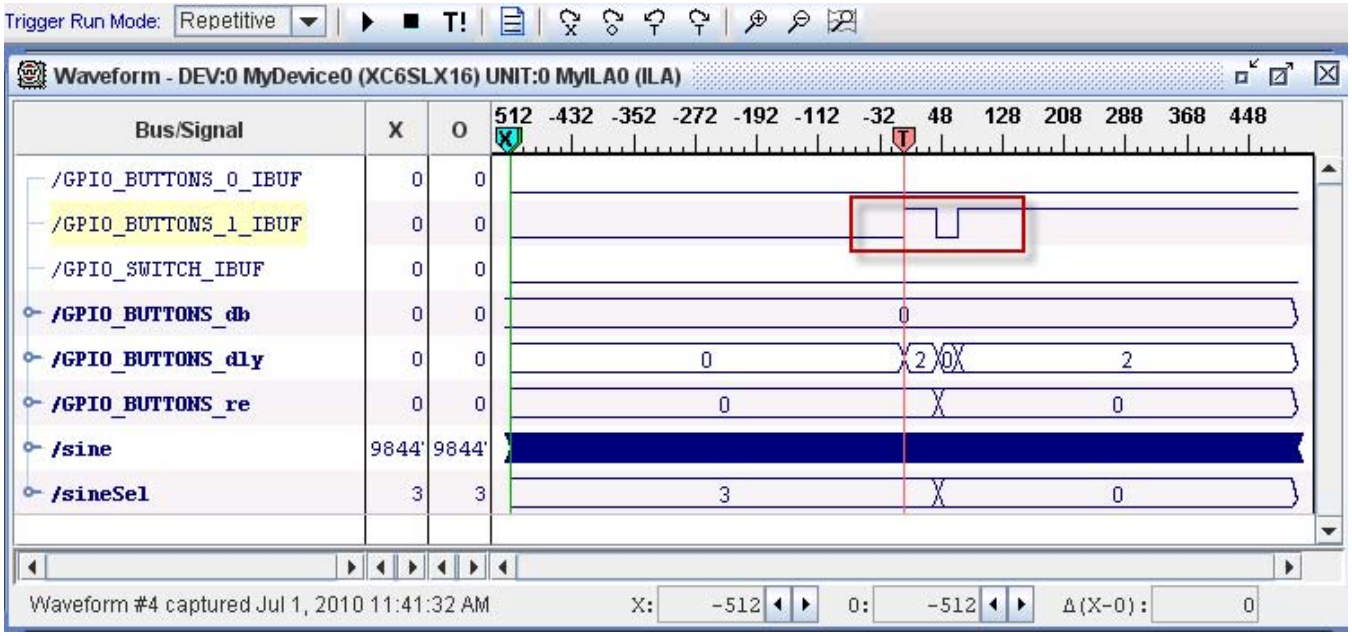


図 10 : GPIO_BUTTONS_1_IBUF 信号のグリッチ

### 信号グリッチの修正と正しいステート マシン動作の確認

ボタンを押すと電気接点が接続されたり解除されたりするので、複数の遷移グリッチまたはバウンスが発生します。この信号バウンスを削除するには、デバウンス回路が必要です。

1. SP601 ボードの DIP スイッチの位置 1 (6 ページの図 1 の Debounce Enable 部分) を ON の位置に設定し、デバウンス回路をイネーブルにします。
2. 上記の手順 6 と 7 を繰り返し、次を確認します。
   - [Sine Wave Sequencer] ボタンを 1 度押したときに複数の遷移が GPIO_BUTTON_re[1] に表示されなくなります。
   - sineSel 信号がボタンを押すごとに 00 → 01 → 10 → 11 に遷移した後 00 に戻ると、ステート マシンが正しく動作しています。

# その他のリソース

## ザイリンクス リソース

- 『ISE® Design Suite : インストールおよびライセンス ガイド』(UG798) : http://japan.xilinx.com/support/documentation/dt_ise13-1_releasenotes_knownissues.htm
- 『ISE Design Suite 13 : リリース ノート ガイド』(UG631) : http://japan.xilinx.com/support/documentation/dt_ise13-1_releasenotes_knownissues.htm
- 『Spartan-6 PCB デザイン ガイド』(UG393) : http://japan.xilinx.com/support/documentation/sw_manuals/xilinx13_1/ug393.pdf
- ザイリンクス資料 : http://japan.xilinx.com/support/documentation
- ザイリンクス用語集 : http://japan.xilinx.com/support/documentation/sw_manuals/glossary.pdf
- ザイリンクス サポート : http://japan.xilinx.com/support

## ChipScope 資料

- 『ChipScope™ Pro ソフトウェアおよびコア ユーザー ガイド』(UG029) : http://japan.xilinx.com/support/documentation/sw_manuals/xilinx13_1/chipscope_pro_sw_cores_ug029.pdf
- 『Using Xilinx ChipScope Pro ILA Core with Project Navigator to Debug FPGA Applications』(UG750) : http://japan.xilinx.com/support/documentation/sw_manuals/xilinx13_1/ug750.pdf

## PlanAhead 資料

- 『PlanAhead™ ユーザー ガイド』(UG632) : http://japan.xilinx.com/support/documentation/sw_manuals/xilinx13_1/PlanAhead_UserGuide.pdf
- 『PlanAhead ソフトウェア チュートリアル : クイック フロー概要』(UG673) : http://japan.xilinx.com/support/documentation/sw_manuals/xilinx13_1/PlanAhead_Tutorial_Quick_Front-to-Back_Overview.pdf

## ボード資料

- Spartan-6 FPGA SP601 評価キットの情報 :http://japan.xilinx.com/products/devkits/EK-S6-SP601-G.htm