

Quartus[®] Prime Signal Tap ロジック・アナライザ 簡易チュートリアル



2018 年 2 月 Rev.1

ELSENA,Inc.





Quartus Prime

Signal Tap ロジック・アナライザ 簡易チュートリアル

<u>目次</u>

| 1. はじめに | .3 |
|-------------------------|----|
| 1-1. 必要な環境 | 3 |
| 1-2. サンプル・デザインの概要 | 4 |
| 2. プロジェクトを開く | .5 |
| 3. Signal Tap IP を作成・追加 | .7 |
| 3-1. STP ファイルを作成・追加 | 7 |
| 3-2. クロックを設定 | 8 |
| 3-3. モニタ信号およびトリガ信号を登録 | 9 |
| 3-4. サンプル容量、バッファ・モードの設定 | ٤4 |
| 3-5. トリガの設定 | ٤4 |
| 4. コンパイルおよびダウンロード | 16 |
| 5. 信号をアナライズ | L7 |
| 改版履歴 | 18 |



1. <u>はじめに</u>

本資料は、Quartus Prime に搭載された Signal Tap ロジック・アナライザ(以下、Signal Tap)機能の操作手順をチュートリアル形式にまとめたものです。本資料と共に提供されているサンプル・デザインを用いて、Signal Tap IP の 作成から実機デバッグ実行までの作業フローを簡易的に理解することができます。

Signal Tap に関しては、このチュートリアルの他に下記資料も併せてご参考ください。

[Design Debugging with the Signal Tap Logic Analyzer]

「Quartus Prime はじめてガイド - Signal Tap ロジック・アナライザの使い方」

Notes: Signal Tap は、Quartus Prime 17.0 以前のバージョンで搭載されていた SignalTap™ II と同じ機能です。 ver.17.1 以降、名称が変更されました。

本チュートリアルは Quartus Prime 17.1 以降の環境でご利用になることを推奨致します。

1-1. 必要な環境

本チュートリアルを実施するためには、以下の環境が必要です。

- Quartus Prime(Lite、Standard、Pro Edition のいずれか)17.1 以降
- インテル[®] FPGA ダウンロード・ケーブル II (旧 USB-Blaster[™] II)[※]
 - インテル FPGA ダウンロード・ケーブル (旧 USB-Blaster)[※] または インテル
 FPGA イーサネット・ケーブル (旧 EthernetBlaster II)でも可

Note: これらケーブルの利用には、専用ドライバのインストールが必要です。

【TIPS】 USB-Blaster™ II のドライバをインストールしてみよう

【TIPS】 USB-Blaster ™ のドライバをインストールしてみよう

- インテル FPGA が搭載されたボード
 - FPGA の JTAG ポートがボード上に引き出され、10pin ヘッダに接続され ているものであれば使用可能です。(ボード上に LED または 7 セグメン ト LED が搭載されていれば、サンプル・デザインを使用しやすいです。)
 - Note: インテル製の FPGA 評価キットや開発キット向けにサンプル・デザイン がいくつか用意されています。対象のボードは、<u>こちらのページ</u>でご確 認ください。
- 本チュートリアル用のサンプル・デザイン
 - <u>こちらのページ</u>からダウンロードしてください。
 - ZIP ファイルの内容をパソコン上の任意のフォルダに解凍します。フォルダのパス、ならびにフォルダ
 <u>名に全角やスペースは使用しないでください</u>。
- 関連情報: <u>「Quartus Prime はじめてガイド Signal Tap ロジック・アナライザの使い方</u>」の"2. 使用環境"をご 参考ください。





1-2. サンプル・デザインの概要



本チュートリアル用のサンプル・デザインのブロック図は以下のとおりです。

LED の点灯仕様は下表を参考にしてください。本チュートリアルで提示した以外のボードをご利用になる場合は、 デコーダ回路を編集するなどして対応してください。

| | | 🗖 点灯 🗖 消灯 |
|-------------|--|--|
| カウンタ 出力値 | Intel® Cyclone® 10 LP FPGA Evaluation Kit | MAX [®] 10 FPGA Evaluation Kit |
| 0 | | |
| 1 | | |
| 2 | | |
| 3 | | |
| 4 | | |
| 5 | | |
| 6 | | |
| 7 | | |
| 8 | | |
| 9 | | |
| | <u>\</u> | |

このデザインは、不具合のある(実機で 10 進カウンタの出力が 9 のときの LED 表示が目視できない)コードで 構成してあります。ここへ SignalTap IP を追加し、内部信号の状態をモニタしましょう。



2. <u>プロジェクトを開く</u>

このチュートリアルのデザイン・ファイルには、すぐに作業が開始できるように指定した評価ボードに対応した Quartus Prime プロジェクト・ファイル(.qsf)、Quartus Prime 制約ファイル(.qsf)、HDL デザインが含まれています。 もしそれ以外のボードをご利用になる場合は、デバイス選択、ピン・レイアウト、各種オプションなどをボードの仕様 に合わせユーザ自身が設定を行ってください。

チュートリアル・プロジェクトを開くには、下記のステップを行います。

- ① Quartus Prime を起動します。
- ② 使用する評価ボードに該当する Quartus Prime プロジェクト・ファイル(st_demo.qpf)ファイルを開きます。
 - a. File メニュー ➤ Open Project をクリックします。
 - b. ZIP ファイル解凍先のフォルダ(作業フォルダ)に移動します。
 - c. st_demo.qpf ファイルを選択し、Open をクリックします。

下図のように、エクスプローラから作業フォルダ内の st_demo.qpf ファイルを Quartus Prime の GUI 上へ ドラッグ&ドロップしても、プロジェクトがセットされます。

Note: 下図は Cyclone 10 LP FPGA 評価キットの場合のフォルダ名です。ダウンロードしたサンプル・デザインにより、フォルダ名は異なります。





すでに、デザイン作成やピン・レイアウトなどの制約設定は完成しています。コンパイルを実行します。

③ Processing メニュー ➤ Start Compilation を選択し、コンパイルを実行します。

FPGA に st_demo.sof をダウンロードして、実機上で動作を確認しましょう。

- ④ ダウンロード・ケーブルの USB ケーブルをパソコンに接続し、ダウンロード・ケーブルのコネクタ部分を評価ボードに装着します。その後、ボードに電源を投入します。
- 5 Programmer (Tools メニュー)を起動し、以下の項目を設定してください。
 - a. Hardware Setup:使用するダウンロード・ケーブルを選択します。
 - b. JTAG モードを選択します。
 - c. プロジェクト・フォルダ内の output_files/st_demo.sof を選択します。(Programmer 起動時に自動的に セットされています。)Device 欄に表示されたデバイスが、評価ボードに搭載された FPGA の型番と同 じであるかを確認してください。
 - d. Program/Configure オプションを有効にします。



6 Start ボタンをクリックし、st_demo.sof ファイルを FPGA にダウンロードします。

ダウンロード完了後、評価ボード上の LED の点灯を確認してください。本チュートリアル 4 ページに示された LED の点灯仕様表にもあるように、10 進カウンタの出力が 9 であるときに表示されるはずの LED が点灯してい ないことがわかります。

3. Signal Tap IP を作成・追加

実機動作をふまえ、なぜカウンタの出力「9」のときの LED が表示されていないのかを Signal Tap で検証して みましょう。

- 3-1. STP ファイルを作成・追加
 - Tools メニュー ➤ Signal Tap Logic Analyzer を選択し、新規 STP ファイルを表示します。
 - ② STP ファイル内の File メニュー ➤ Save As... において、保存するフォルダおよびファイル名(任意)を指定します。このチュートリアルでは、下記の環境でファイルを保存してください。
 - ・保存フォルダ: Quartus Prime のプロジェクト・フォルダの直下
 - ・保存ファイル名: st_demo.stp

Add file to current project オプションを有効にして "保存(S)" ボタンをクリックします。

Input "Data and Trigger" is empty とメッセージ・ボックスが表示されるので、OK ボタンをクリックします。

| 🕥 Quar | tus Prime 🔀 |
|--------|-----------------------------------|
| 0 | Input "Data and Trigger" is empty |
| | ОК |

③ Do you want to enable Signal Tap File "st_demo.stp" for the current project? とメッセージが表示されます。 Yes ボタンをクリックし、STP ファイルをプロジェクトに登録します。

| 🕥 Quar | tus Prime |
|--------|--|
| | Do you want to enable Signal Tap File "st_demo.stp" for the current project? |
| | Yes No Cancel |

関連情報: <u>「Quartus Prime はじめてガイド - Signal Tap ロジック・アナライザの使い方</u>」の "4-1. デザインに Signal Tap を追加' をご参考ください。



3-2. クロックを設定

Signal Tap IP のクロックを指定します。

- STP ファイル(Setup タブ)の Signal Configuration ペインにおいて、Clock 欄右のブラウズ・ボタンをクリック し、Node Finder を表示させます。
- ② Node Finder の Filter を Signal Tap: post-fitting に切り替え、List ボタンをクリックします。
- ③ 検出された Matching Nodes リストの信号から clk~inputclkctrl をダブルクリックで指定し、右側の Nodes Found リストに選出します。

| rigger: | THESE #1 | Lock mode | : 🥂 Allow all chi | anges 🔹 | Signal Configuration: | | |
|--|--|--|--|---|---|------------|----------|
| | Node | Data Enab | e Trigger Enable | e Trigger Conditions | | | 1 |
| ype Alias | Name | 0 | 0 | 1 🗹 Basic AND 🔻 | Clock clk~inputclkctrl | | |
| ardela altalezza ada | d d | | | | Dete | | |
| Mode Finder | | | | | | | F |
| | | | | | | type: Auto | |
| Named: * | | | | | ▼ List 🔍 | gments | ÷ |
| Ontions | | | | | | 0 | E 17 |
| Filter Sin | nal Tao: port-fittio | 10 | | | Customize | Manual: | 0 |
| | | | | | | | |
| | nai rap. post num | '9 | | | Contoniazio | | - |
| Look in: st | _demo | 2 | | V | Include subentities 💟 Hierarchy view | | • |
| Look in: st | _demo | 5 | 121 (21 | Voder Found: | Include subentities V Hierarchy view | | • |
| Look in: st | _demo les: | Turs | e e | Vodes Found: | Include subentities V Hierarchy view | | • • |
| Look in: st | _demo les: Name | Туре | ∰i ∰i Creator | Nodes Found: | Include subentities V Hierarchy view | | |
| Look in: st Matching Nod st_demo | _demo es: Name | Type | Creator + | ▼ ♥ Nodes Found: Name Clk~inputclkctrl | Include subentities V Hierarchy view e Assignments Unassigned | | |
| Look in: st. Matching Nod st_demo | _demo es: Name t | Type Input Combinational | Creator User entered | Nodes Found: Nodes Found: Name karinputciketri | Include subentities V Hierarchy view e Assignments Unassigned | | • |
| Look in: st, Matching Nod st_demo | _demo les: Name t t∼input | Type Input Combinational Input | Creator User entered Compirated User entered | Nodes Found: Name koninguteiketri | Include subentities V Hierarchy view e Assignments Unassigned | | • |
| Look in: st, Matching Nod st_demo in_areset areset in_clk | _demo] ies: Name t∼input nput | Type Input Combinational Input Combinational | Creator User entered Compirated User entered Compirated | Nodes Found: Name k~inputcikctri | Include subentities V Hierarchy view e Assignments Unassigned | | • |
| Look in: st Matching Nod st_demo in_areset areset areset c_areset c c_areset c_areset c_areset c_areset c_areset c_areset c_areset c_areset c_areset c_areset c_areset c_areset c_areset c_areset c_areset c_areset c_areset c c_areset c_areset c_areset c_areset c c_areset c c_areset c c_areset c c c c c c c c c c c c c c c c c c c | _demo ies: Name t~input nputtktrl | Type Input Combinational Input Combinational Combinational | Creator | Nodes Found: Name ck~inputckctrl | Include subentities V Hierarchy view e Assignments Unassigned | | • |
| Look in: st. Matching Nod st_demo st_demo in: areset in: dk- clk-ir it clk-ir it clk-ir it clk-ir | _demo] ies: Name t t~input nputcikctrl III~U | Type Input Combinational Input Combinational Combinational Combinational | Greator Creator User entered Compirated User entered Compirated Compirated Compirated | Nodes Found: Name Ck~inputckctrl | Include subentities V Hierarchy view e Assignments Unassigned | | • • |

Note: Signal Tap IP のクロックに用いる信号は、検証の精度を向上するためにトリガに用いる信号やサンプリン グしたい信号と同期の取れたクロック・ドメインを指定することを推奨します。

- ④ OK ボタンをクリックし、Signal Tap IP のクロックを登録します。
- 関連情報: <u>「Quartus Prime はじめてガイド Signal Tap ロジック・アナライザの使い方」</u>の"4-2-1. クロックの設 定"をご参考ください。

3-3. モニタ信号およびトリガ信号を登録

カウンタの出力「9」のときの LED が表示されていない原因を探すため、以下の出力信号をモニタします。

- ・ 10 進アップ・カウンタ回路の出力 [信号 A]
- ・ デコーダ回路の出力 [信号 B]



信号 A および信号 B を STP ファイル上に登録するため、下記のステップを行います。

① STP ファイルの Node List の空白部分をダブルクリックし、Node Finder を起動します。

| Instance Manager: 📉 🕟 🔳 🛅 Invalid JTAG conf | figuration | |
|---|--|---|
| Instance Status Er | 🕫 Node Finder | |
| 🛃 auto_signaltap_0 Not running 🛛 🛛 | | |
| | Named: * | - List |
| < | Options | |
| | Filter: Signal Tap: post-fitting | Customize |
| trigger: #1 | Look in: [st_demo] | Include subentities Hierarchy view |
| Node | Matching Nodes: BI BI No | odes Found: |
| Type Alias Name | Name Type Creat 🕬 | Name Assignments |
| ダブルクリック | < < < < < < < < < < < < < < < < < < | Insert Close |

② Node Finder ウィンドウ内 Filter 欄のプルダウン・リストにおいて、Signal Tap: post-fitting を選択します。

| 🦇 Node Find | ler | | X |
|-------------|--------------------------|-------------|----------------|
| Named: | * | • | List 🔝 |
| Filter: | Signal Tap: post-fitting | • | Customize |
| Look in: | st_demo | bentities 🔽 | Hierarchy view |



③ まずは、信号 A (10 進アップ・カウンタ回路の出力)を選出しましょう。

下位階層 bcounter (10 進アップ・カウンタ回路)の出力信号 q (4bit)をドライブする 4bit レジスタ cnt を 検索しやすくするため、Look in 欄の右にあるボタンをクリックし、Select Hierarchy Level ダイアログ・ボック スから bcounter:u2 をハイライト選択し、OK ボタンをクリックします。

| Options | | | | | | | |
|---------------------------|--|--|--|--|--|--|--|
| Filter: Signal Tap: post- | Signal Tap: post-fitting Customize | | | | | | |
| Look in: st_demo | ▼ ▼ Include subentities ▼ Hierarchy view | | | | | | |
| Matching Nodes: | 🥐 Select Hierarchy Level | | | | | | |
| Name 1 | Assignments | | | | | | |
| | | | | | | | |
| | Entity:Instance | | | | | | |
| | | | | | | | |
| | abo prescaler.u1 | | | | | | |
| | abe bcounter.u2 | | | | | | |
| | abo WHD deco:u3 | | | | | | |
| ٠ III | • | | | | | | |
| | OK Cancel Help rt Close | | | | | | |

- ④ Node Finder の List ボタンをクリックすると、bcounter:u2 インスタンス内の信号が Matching Nodes リスト (左枠)に検出されます。
- ⑤ Matching Nodes リストから該当するレジスタ(cnt[0]、cnt[1]、cnt[2]、cnt[3])をダブルクリックで選択し、 Nodes Found リスト(右枠)に登録して、Insert ボタンをクリックします。(Close ボタンで Node Finder を閉じ ます。)

| Named: * |
|---|
| Matching Nodes: Image: Creator Nodes Found: Name Type Creator Name Assignments bcounter:u2 Add0~0 Comnal Compirati Image: Creator Image: Creator Add0~1 Comnal Compirati Image: Creator Image: Creator |



STP ファイルの Node List に内部信号が登録されます。

| trigger: #1 Lock mode: Allow all changes | | | | | | |
|--|-------|----------------------|-------------|----------------|--------------------|-------------------|
| Node | | | Data Enable | Trigger Enable | Trigger Conditions | |
| Туре | Alias | Name | | 4 | 4 | 1 🗹 Basic AND 🛛 🔻 |
| R | | bcounter:u2 cnt[3] 🔫 | | V | V | |
| R | | bcounter:u2 cnt[2] | 「旧来友」 | 入り基づ可能 | V | |
| R | | bcounter:u2 cnt[1] | 「原田で、 | | V | |
| R. | | bcounter:u2 cnt[0] | | V | V | |

Note: Matching Nodes リストから選出する順番が、STP ファイルに登録される上下の順番になります。上 下の順番は、Node Finder で選出後も STP ファイル内でドラッグ&ドロップにより入れ替えることが 可能です。

これらのレジスタを、キーボードの Shift キーを使用して全選択し、右クリック ➤ Group によりバス表示に 変更します。



一旦 STP ファイルを保存しておきます。(File メニュー ➤ Save)

⑥ 続いて、信号 B (デコーダ回路の出力)を選出します。この信号は、Technology Map Viewer (post-Fitting)
 を活用して視覚的に検索し、選出してみましょう。

Quartus Prime 上の Tools メニュー ➤ Netlist Viewers ➤ Technology Map Viewer (Post-Fitting) によりビュー ワを起動します。



- Note: Technology Map Viewer は、自分のデザインの論理合成前および配置配線後のネットリストがグラフ ィカルに表示されるビューワで、Post-Mapping と Post-Fitting の2 タイプが用意されています。今回 Signal Tap で活用するのは Post-Fitting のビューワです。Netlist Navigator (ウィンドウの左側)には プロジェクト・デザインが階層表示されていて、各インスタンスおよびそのインスタンスで使用されて るポートやロジックが表示されています。これらのアイテムから、Signal Tap でモニタ信号およびトリ ガ信号として使用する信号を探します。
- ⑦ Netlist Navigator から deco:u3 (デコーダ回路)を探し、deco:u3 ⇒ Primitives ⇒ Atoms と下位を展開します。Atoms 内のノードをどれか一つダブルクリックすると、指定した階層のノードがグラフィカル・ビュー 側に表示されます。

ここから、内部ノード Mux*~0 を選択します。(キーボードの Ctrl キーを押しながらマウスでノードを選択 することで、複数選択が可能です。)

⑧ 指定したノードをハイライトさせたままグラフィック・ビュー側にマウスを移動させ、右クリック ➤ Add Node to Signal Tap Logic Analyzer ➤ st_demo.stp を選択します。



※ もし、New Signal Tap File しか表示されない場合は、st_demo.stp ファイルが閉じています。st_demo.stp ファイルを開いた 後に、Technology Map Viewer (post-Fitting) を使用してください。

STP ファイルの Node List にノードがインポートされたことを確認します。

関連情報: 【TIPS】 SignalTap® II でキャプチャしたい内部信号を簡単に STP ファイルに登録する方法



④ deco:u3|Mux*~0 を bcounter:u2|cnt[3..0] 同様に、キーボードの Shift キーを使用して全選択し、右クリック ➤ Group によりバス表示に変更します。また、グループ化した信号を選択し、右クリック ➤ Rename により、グループ名 deco:u3|Mux を付けます。



- 10 STP ファイルを保存しておきます。(File メニュー ► Save)
- Note: Quartus Prime の Project Navigator 上からインスタンスを選択し、Technology Map Viewer にクロスプロ ーブすることも可能です。



関連情報: <u>「Quartus Prime はじめてガイド - Signal Tap ロジック・アナライザの使い方」</u>の"4-2-2. 信号の登録" をご参考ください。

3-4. サンプル容量、バッファ・モードの設定

Signal Configuration ペイン内 Data エリアの設定をします。

- ① Sample depth を 1K に設定します。
- ② Storage qualifier の Type を Continuous に設定します。

| Signal Configuration: |
|--------------------------------------|
| Clock: clk~inputclkctrl |
| Data |
| Sample depth: 1 K 🔹 RAM type: Auto 💌 |
| Segmented: 2 512 sample segments |
| Nodes Allocated: Auto Manual: 9 |
| Pipeline Factor: 0 |
| Storage qualifier: |
| Type: Continuous |

関連情報 「Quartus Prime はじめてガイド - Signal Tap ロジック・アナライザの使い方」の "4-2-3. サンプル容 量の指定"、"4-2-4. RAM タイプの指定"、"4-2-5. バッファ・モードの選択" をご参考ください。

3-5. トリガの設定

Signal Configuration ペイン内 Trigger エリアの設定をします。

- ① Trigger position を Center trigger position に設定します。
- ② Trigger conditions を 3 に設定します。
- ③ それ以外の設定はデフォルトのままにします。

| Trigger | | | | |
|-----------------------|------------|-----------------|---|---|
| Nodes Allocated: () | Auto | 🔘 Manual: | 5 | × |
| Trigger flow control: | Sequential | | | • |
| Trigger position: | 🕸 Center t | rigger position | | - |
| Trigger conditions: | 3 | | | - |

④ グループ名 bcounter:u2|cnt[3..0] のトリガ・パターンを設定します。

"LED にカウンタの値が 9 の際の動作が見られないのは、カウンタが 9 を出力していないからか?"と 想定し、カウンタ出力が 8、9、0 の間をモニタするトリガ条件を設定します。



Node List 上で bcounter:u2|cnt[3.0] の Trigger Conditions 1 欄を右クリックで選択し、Insert Value を選択 します。Insert Value ウィンドウにおいて、Radix を Hexadecimal に切り替え、value 欄に 8 を入力し、OK ボタンをクリックします。

| Node | Data Enable | | Trigger Conditions | | | | | | |
|--|---|--|--|--|--|--|--|--|--|
| Name | 9 | 9 | 1 🗹 Basic ANI | D 🔻 | 2 🗹 Basic AND 🔻 | 3 🗹 Basic ANE 🔻 | | | |
| 🖵 bcounter:u2 cnt[30] | V | V | Xh | > | xb | Xh | | | |
| bcounter:u2 cnt[3] | V | | | | AND / OR | | | | |
| bcounter:u2 cnt[2] | V | | | | | | | | |
| bcounter:u2 cnt[1] | V | \checkmark | | | AND | | | | |
| bcounter:u2 cnt[0] | V | V | | | OR | | | | |
| 📮 deco:u3 Mux3 | V | | Xh | | NAND | Xh | | | |
| deco:u3 Mux3~0 | V | V | | | NOR | | | | |
| deco:u3 Mux2~0 | V | V | | | Non | | | | |
| deco:u3 Mux1~0 | V | V | | | XOR | | | | |
| deco:u3 Mux0~0 | V | V | | | XNOR | | | | |
| isert Value ue: 8 Radix: Hexa OK | decimal | ncel Help | | ■ 2 √ 7 X | FALSE Compare Don't Care Low Falling Edge Rising Edge High Either Edge | | | | |
| | Node Name Name bcounter:u2[cnt[3.0] bcounter:u2[cnt[3] bcounter:u2[cnt[2] bcounter:u2[cnt[1] bcounter:u2[cnt[0] deco:u3]Mux3~0 deco:u3]Mux3~0 deco:u3]Mux2~0 deco:u3]Mux1~0 deco:u3]Mux0~0 sert Value sert Value OK | Node Data Enable Name 9 bcounter:u2]cnt[3.0] IV bcounter:u2]cnt[3] IV bcounter:u2]cnt[2] IV bcounter:u2]cnt[1] IV bcounter:u2]cnt[0] IV bcounter:u2]cnt[0] IV bcounter:u2]cnt[1] IV bcounter:u2]cnt[0] IV bcounter:u2]cnt[0] IV cdeco:u3]Mux3~0 IV deco:u3]Mux2~0 IV deco:u3]Mux0~0 IV deco:u3]Mux0~0 IV | Node Data Enable Trigger Enable Name 9 9 > bcounter.u2]cnt[3.0] V V > bcounter.u2]cnt[3] V V > bcounter.u2]cnt[2] V V > bcounter.u2]cnt[1] V V > bcounter.u2]cnt[0] V V > bcounter.u2]cnt[0] V V > bcounter.u2]cnt[0] V V > deco:u3]Mux3 V V - deco:u3]Mux3~0 V V - deco:u3]Mux2~0 V V - deco:u3]Mux0~0 V V - deco:u3]Mux0~0 V V | Node Data Enable Trigger Enable Name 9 9 1 Basic AN bcounter.u2[cnt[3.0] V V Xh bcounter.u2[cnt[3] V V Xh bcounter.u2[cnt[1] V V Xh bcounter.u2[cnt[1] V V Xh bcounter.u2[cnt[0] V V Xh cdeco:u3[Mux3~O V V Xh deco:u3[Mux2~O V V Xi deco:u3[Mux0~O V V Xi sert Value X Xi Xi Max Max Ki Ki Max Max Ki Ki Max Ki Ki Ki | Node Data Enable Trigger Enable Name 9 9 1 Basic AND - bcounter.u2[cnt[30] V V Xh - bcounter.u2[cnt[2] V V Xh - bcounter.u2[cnt[1] V V Xh - bcounter.u2[cnt[1] V V Xh - bcounter.u2[cnt[0] V V Xh - deco:u3[Mux3~0 V V Xh - deco:u3[Mux2~0 V V Xh - deco:u3[Mux0~0 V V Xh sert Value X Image: Set Value Image: Set Value Je: 8 Radix: Hexadecimal Image: Set Value | Node Data Enable Trigger Enable Trigger Conditions Name 9 9 1 Basic AND ▼ 2 Basic AND ▼ > bcounter.u2 cnt[3.0] 0 0 ×h ×h > bcounter.u2 cnt[3] 0 0 ×h AND / OR > bcounter.u2 cnt[1] 0 0 0 AND / OR > bcounter.u2 cnt[0] 0 0 0 0 0 > deco:u3 Mux3-0 0 0 0 0 0 0 > deco:u3 Mux0-0 0 0 0 0 0 0 0 gent deco:u3 Mux0-0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | | | |

同様に bcounter:u2|cnt[3..0] の Trigger Conditions 2、Trigger Conditions 3 欄をダブルクリックし、9、0 を順 に入力します。

| | | Node | Data Enable | Trigger Enable | Trigger Conditions | | | | |
|----------|-------|---|-------------|----------------|--------------------|-----------------|-----------------|--|--|
| Туре | Alias | Name | 9 | 9 | 1 🗹 Basic AND 🔻 | 2 🗹 Basic AND 🔻 | 3 🗹 Basic ANE 🔻 | | |
| - Barris | | ^[] bcounter:u2 cnt[30] | V | V | 8h | 9h | Oh | | |
| * | | bcounter:u2 cnt[3] | V | \checkmark | 1 | 1 | 0 | | |
| * | | bcounter:u2 cnt[2] | V | V | 0 | 0 | 0 | | |
| * | | bcounter:u2 cnt[1] | V | V | 0 | <u>0</u> | 0 | | |
| * | | bcounter:u2 cnt[0] | V | V | <u>0</u> | 1 | 0 | | |
| - | | ⊡ deco:u3 Mux3 | V | | Xh | Xh | Xh | | |

⑤ グループ名 deco:u3|Mux のノード名部分を右クリック選択 ➤ Bus Display Format ➤ Binary を選択し、 Radix をバイナリに変更します。





また、deco:u3|Mux はトリガ信号として用いないため、Trigger Enable オプションを OFF にします。

| | | Node | Data Enable | Trigger Enable | e Trigger Conditions | | | | |
|------|-------|----------------------------|-------------|----------------|----------------------|-----------------|-----------------|--|--|
| Туре | Alias | Name | 9 | 5 | 1 🗹 Basic AND 🔻 | 2 🗹 Basic AND 🔻 | 3 🗹 Basic ANE 🔻 | | |
| | | ₽bcounter:u2 cnt[30] | V | V | 8h | 9h | Oh | | |
| * | | bcounter:u2 cnt[3] | V | V | 1 | 1 | 0 | | |
| * | | bcounter:u2 cnt[2] | v | V | 0 | 0 | 0 | | |
| * | | bcounter:u2 cnt[1] | V | V | 0 | 0 | 0 | | |
| * | | bcounter:u2 cnt[0] | V | V | 0 | 1 | 0 | | |
| - | | ₽ deco:u3 Mux3 | V | | | | | | |
| * | | deco:u3 Mux3~0 | v | | | | | | |
| * | | deco:u3 Mux2~0 | V | | | | | | |
| * | | deco:u3 Mux1~0 | v | | | | | | |
| * | | deco:u3 Mux0~0 | V | | | | | | |

⑥ STP ファイルを保存しておきます。(File メニュー ► Save)

4. <u>コンパイルおよびダウンロード</u>

① STP ファイルの Processing メニュー ➤ Start Compilation または Start Rapid Recompile を選択し、Signal Tap ロジック・アナライザ IP を組み込んだユーザ・デザインをコンパイルします。



Notes: Rapid Recompile は、該当する FPGA ファミリのみ実行可能です。

- sof ファイルをダウンロードするため、事前にパソコンとダウンロード・ケーブル、ダウンロード・ケーブルと FPGA の搭載されたボードを接続し、ボードの電源を投入します。
- ③ STP ファイル内 JTAG Chain Configuration ペインの Hardware 欄において、プルダウン・リスト または Setup ボタンをクリックし、使用するダウンロード・ケーブルを選択します。
- ④ Device 欄に対象の FPGA が表示されていない場合は、Scan Chain ボタンをクリックしてください。
- ⑤ SOF Manager 右端のブラウズ・ボタンをクリックし、"プロジェクト・フォルダ/output_files/st_demo.sof"を選 択します。

関連情報: 「Quartus Prime はじめてガイド - Signal Tap ロジック・アナライザの使い方」の"4-3. トリガの定義" をご参考ください。



⑥ SOF Manager 中央にある Program Device ボタンをクリックし、ダウンロードを開始します。

| JTAG Chai | n Configuration: JTAG ready | × |
|-----------|--|-----------|
| Hardware: | USB-BlasterII [USB-1] | Setup |
| Device: | @1: 10CL025(Y Z)/EP3C25/EP4CE22 (0x020F30DD) | can Chain |
| >> SOF | Manager | 0.50f |
| | Program Device ボタン | |

関連情報: <u>「Quartus Prime はじめてガイド - Signal Tap ロジック・アナライザの使い方」</u>の"4-4. デザインのコン パイル"、"4-5. FPGA プログラミング"をご参考ください。

5. <u>信号をアナライズ</u>

① 解析を実行するため、STP ファイルの Run Analysis ボタンをクリックします。

| <u>F</u> ile | <u>E</u> dit | <u>V</u> iew | <u>P</u> roject | P <u>r</u> ocessing | <u>T</u> ools | <u>W</u> indow | <u>H</u> e |
|--------------|--------------|--------------|-----------------|---------------------|---------------|----------------|------------|
| | | ? (* | 蒜 👪 | ▶ 😫 🛛 | | | |
| Insta | nce Ma | nager: | 3 | 🔳 🛄 Re | ady to a | cquire | |
| Instan | ice | | | Status | | Enabled | |

② 取得したデータを解析します。

すべてのトリガ条件が発生しました。波形をよく見ると、カウンタ出力 bcounter:u2|cnt[3..0] が Oh にリセットされる前に、9h になっている状態があることが確認できます。(Data タブの波形上で左右クリックすること でズームイン、ズームアウトができます。)

| Туре | Alias | Name | - 6 | -4 | -2 | . 9 | ą | | 4 | | ę | |
|------|-------|----------------------------|----------------|-------|----|---------|---|--|-----|-----|---|--|
| - | | □ bcounter:u2 cnt[30] | | 8h | | | | | 0 | h | | |
| * | | bcounter:u2 cnt[3] | | | | | | | | | | |
| * | | bcounter:u2 cnt[2] | | | | | | | | | | |
| * | | bcounter:u2 cnt[1] | | | | | | | | | | |
| * | | bcounter:u2 cnt[0] | | | | | | | | | | |
| - | | Ģ deco:u3 Mux3 | | 0111b | | (1111b) | | | 000 | 00b | | |
| * | | deco:u3 Mux3~0 | | | | | | | | | | |
| * | | deco:u3 Mux2~0 | | | | | | | | | | |
| * | | deco:u3 Mux1~0 | | | | | | | | | | |
| * | | deco:u3 Mux0~0 | | | | | | | | | | |

9h は他の信号と比べて継続時間が短すぎるため、LED に表示されないのだと分析できます。リセットの仕様がおかしいのかもしれません。ソース・コードを見て、修正してみましょう。

関連情報: <u>「Quartus Prime はじめてガイド - Signal Tap ロジック・アナライザの使い方」</u>の"4-6. Signal Tap の実 行"、"4-7. キャプチャされたデータの表示・解析"をご参考ください。



<u> 改版履歴</u>

| Revision | 年月 | 概要 |
|----------|------------|----|
| 1 | 2018 年 2 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.ip/

 株式会社マクニカ アルティマ カンパニー
 https://www.alt.macnica.co.ip/

 株式会社エルセナ
 https://www.elsena.co.ip/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。