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

Ver.17.1

2018 年 1 月 Rev.1

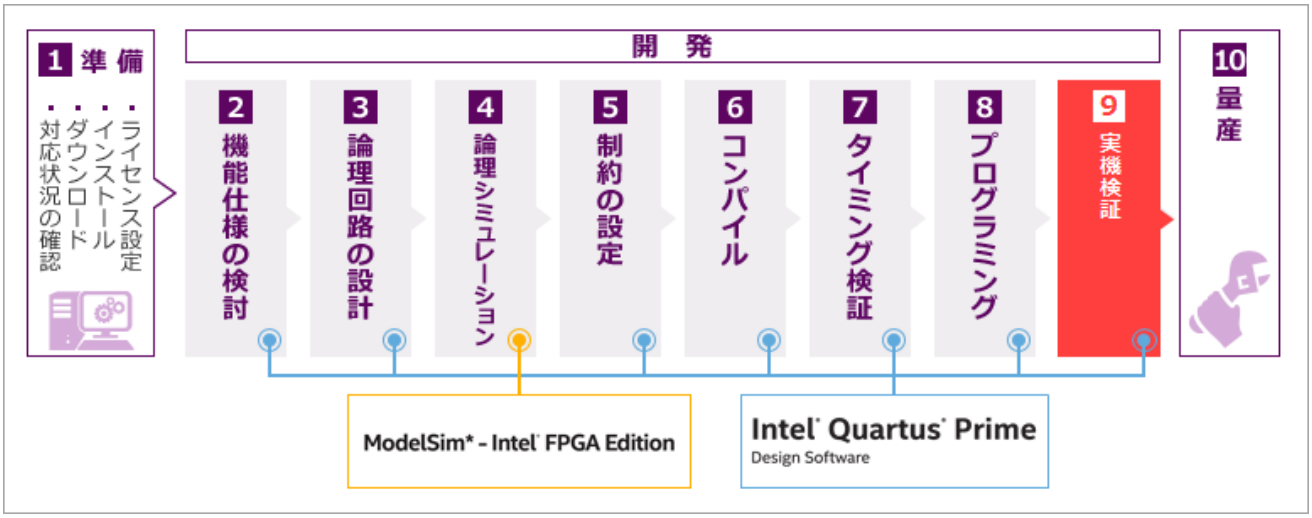
Quartus Prime はじめてガイド
Signal Tap ロジック・アナライザの使い方

目次

1. はじめに

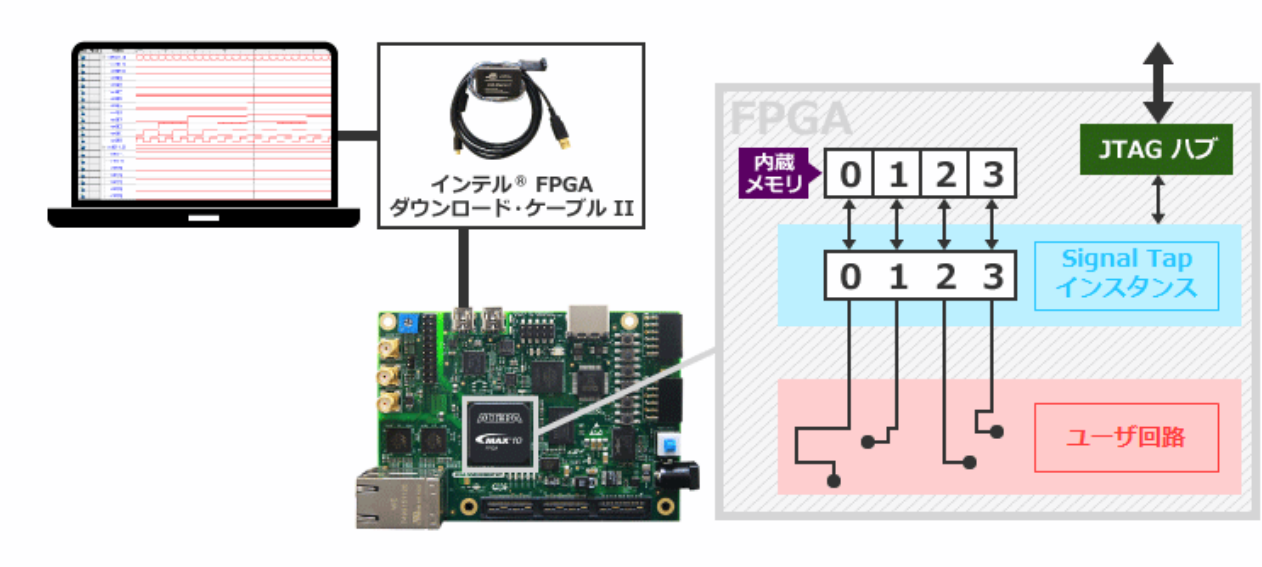
この「Quartus Prime はじめてガイド」シリーズは、Quartus® Prime 開発ソフトウェア（以下、Quartus Prime）をはじめてご利用になるユーザ向けの資料です。

本資料は、FPGA を開発するフローの中で、主に下図の赤枠内の開発フェーズで参考になります。



現在の FPGA は大規模で高性能となっている反面、パッケージや未使用 I/O ピンの制限などにより、デバッグは困難になっています。インテルではその現状を緩和するために、デザインを FPGA 上で動作させながら未使用の I/O ピンを活用することなく内部信号の動作を検証できるオンチップ・デバッグ環境 “Signal Tap ロジック・アナライザ”（以下、Signal Tap）を提供しています。

この Signal Tap 機能は、Quartus Prime に標準搭載されたオンチップ・デバッグ・ツールで、ロジック・アナラーザやオシロスコープなどの外部測定装置を使用せずに、ボード上で動作する FPGA の内部信号の状況をキャプチャおよび表示することができます。



無償の Signal Tap 用 IP コアをユーザ・デザインに組み込みデバイス内部に配置配線し、データをプログラミング後ボード上で動作させます。キャプチャされたデータはデバイスの内部メモリ領域に一度格納され、その後、ボード上の JTAG ピンから通信ケーブルを経由して Quartus Prime に転送され、パソコンの画面上に波形表示

されます。Signal Tap は、1 つの JTAG チェーン内に複数の FPGA デバイスを含んだボード環境でも実行できます。

詳しくは、下記ドキュメントをご覧ください。

「Design Debugging with the Signal Tap Logic Analyzer」(Quartus Prime Standard Edition)

「Design Debugging with the Signal Tap Logic Analyzer」(Quartus Prime Pro Edition)

なお この 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 より名称が変更されました。

2. 使用環境

Signal Tap を使用するには、以下の環境が必要です。

2-1. 開発ソフトウェア

対応する開発ソフトウェアは以下のとおりです。

- Quartus Prime Lite Edition
- Quartus Prime Standard Edition
- Quartus Prime Pro Edition

※ Quartus Prime Lite Edition の場合は、16.0 以前のバージョンを使用する際 TalkBack 機能を有効にする必要があります。詳細はこちらの TIPS をご覧ください。

※ Quartus Prime Programmer and Tools の場合は、解析のみ可能です。デバッグ用デザインの作成やコンパイルなどは行えません。

2-2. 通信ケーブル

対応する通信ケーブルは以下のとおりです。

- 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）
- インテル FPGA イーサネット・ケーブル （旧 EthernetBlaster II）

※ ダウンロード・ケーブルがオンボード（内蔵）タイプの評価ボードを使用する場合は、これらの通信ケーブルは不要です。

2-3. 対応デバイス

対応するデバイスは以下のとおりです。

- Stratix® シリーズ
- Arria® シリーズ
- Cyclone® シリーズ
- MAX® 10

※ 内蔵メモリを持たない CPLD デバイス（MAX® V など）は Signal Tap 機能を使用できません。

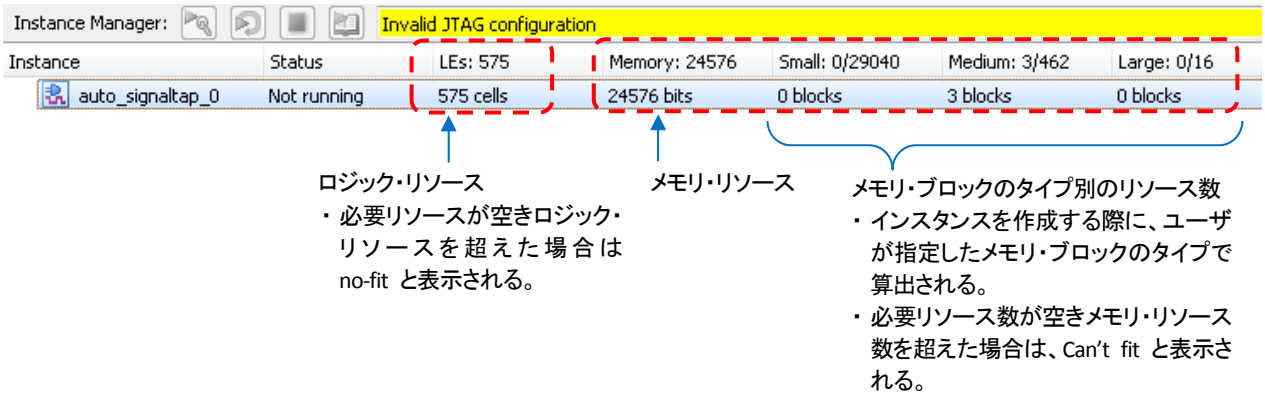
※ 通信用として、ボード上に 10 pin コネクタを用意し、FPGA の JTAG ポートを接続しておく必要があります。

3. 概要

3-1. 必要な FPGA の内部リソース

Signal Tap でデバッグするためには、FPGA 内部に、ロジック・アナライザ・メガファンクションを構築するための “ロジック・リソース” と、キャプチャしたデータを格納するための “内部メモリ・ブロック” が必要です。そのため、ユーザ回路をインプリメントした残りの空きリソースがどのくらいあるかによって、構成できる Signal Tap の仕様、キャプチャできる信号本数やサンプリング数が変わります。

Quartus Prime の Signal Tap ロジック・アナライザ・ファイル (*.stp) には、構築した Signal Tap ファンクションがどのくらいのロジック・リソースとメモリ・ブロックを必要とするかを自動的に推計する機能が備わっており、Instance Manager ペイン内（STP ファイルの左上）で確認ができますので、この参考値を確認してください。なお、実際のリソース使用率と 10% 程度異なる場合があります。



3-2. 観測できない信号

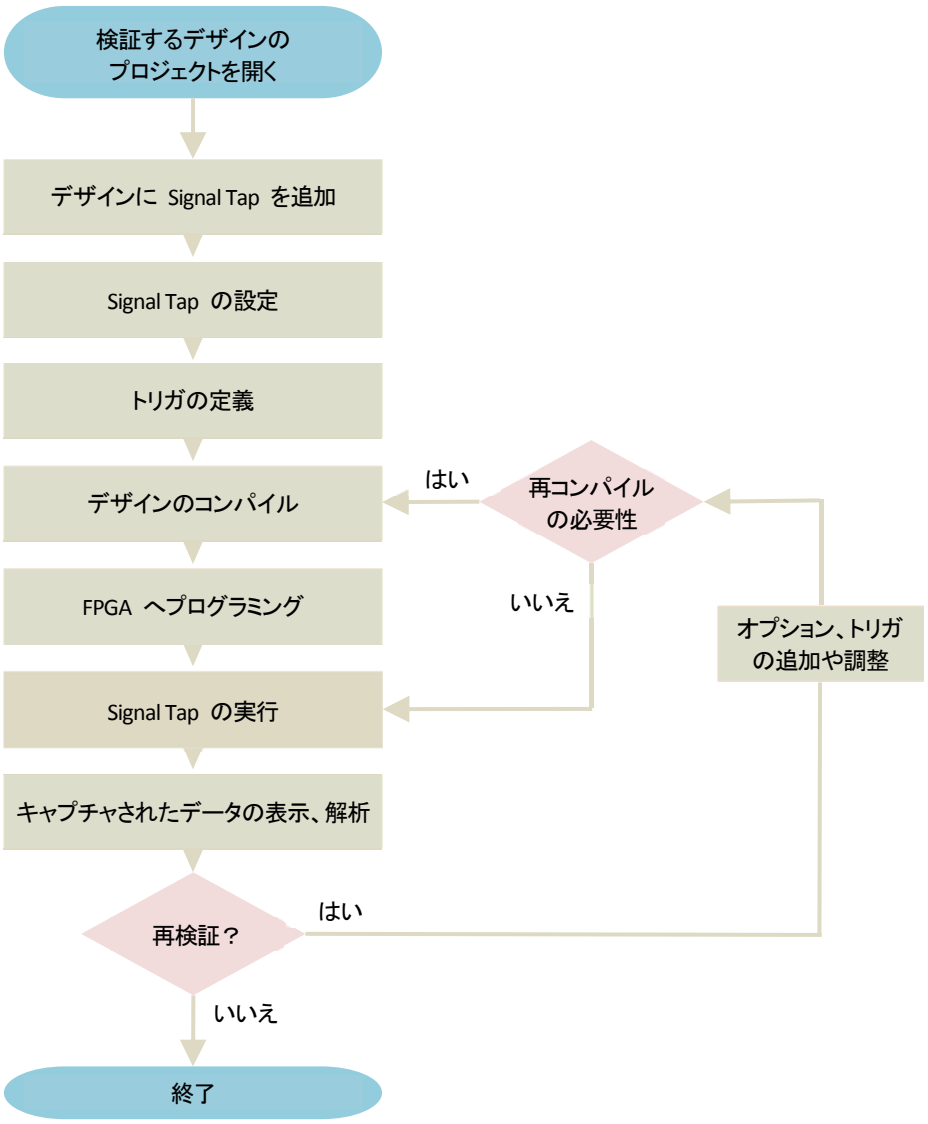
デザインの配置配線後 (post-fitting) に存在する信号すべてが Signal Tap で観測できるわけではありません。以下の信号は観測することはできません。

- JTAG (TCK、TDI、TDO、TMS) コントロール信号
- キャリーチェーンのキャリーアウト信号
- 配置配線後の出力ピン（双方向ピンを含む）

  ～ 出力ピンの信号を観測するには、出力ピンを駆動するレジスタまたはバッファをタップします。
- ALTGXB IP コアのインスタンシエーションの任意ポート
- LVDS のシリアライザ/デシリアライザ (SERDES) ブロックからのデータ出力
- DDR / DDR2 デザインの DQ、DQS 信号

3-3. 作業フロー

Signal Tap の作業フローは以下のとおりです。



3-4. 検証をする前に

Signal Tap はユーザ・デザインの内部信号を観測するには、事前に観測用のトリガとして用いたい信号の指定と、観測したい内部信号の指定を行います。そのため、最低限 Analysis and Elaboration を実行している必要があります。（推奨はフル・コンパイルを実行）

Analysis and Elaboration は以下のメニューで実行します。

Processing メニュー ➤ Start ➤ Start Analysis and Elaboration

ただし、インクリメンタル・コンパイルやラピッド・リコンパイルを採用したコンパイルを実行しているプロジェクトを Signal Tap で検証する場合には、該当するコンパイル手法で実行を完了しておいてください。

4. 操作手順

4-1. デザインに Signal Tap を追加

Signal Tap の IP コアは、ユーザ・ロジックと共にデバイスへ実装させるため、ユーザ・デザインに追加する必要があります。追加方法は以下の 2 通りです。

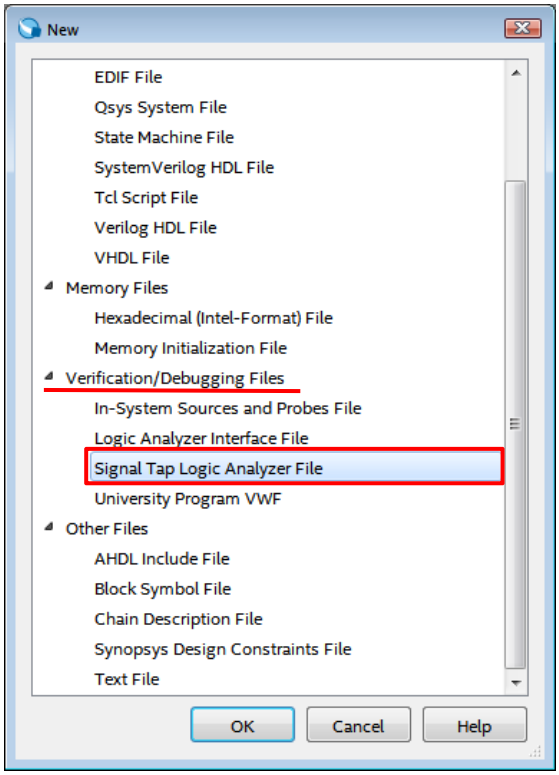
- Signal Tap ファイル (*.stp) を作成し、ユーザ・デザインに自動でインスタンスする
- IP Catalog から Altera SignalTap II Logic Analyzer を選択および作成し、ユーザ・デザインに手動でインスタンスする

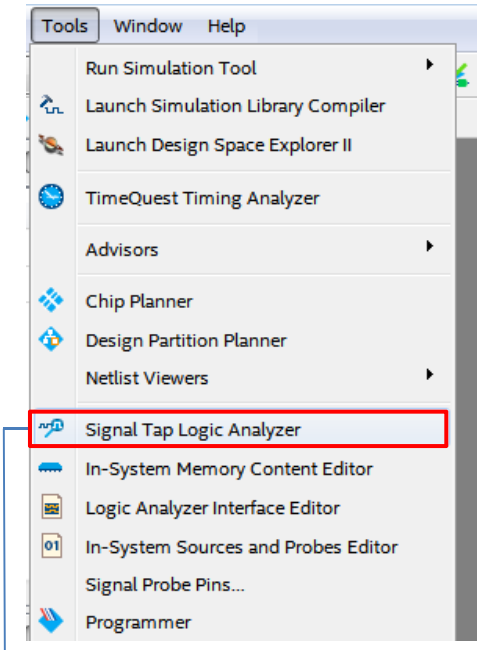
この資料では、Signal Tap ファイル (以下、STP ファイル) を使用する方法をご紹介します。

4-1-1. 新規 STP ファイルを起動

File メニュー ➤ New をクリックし、Verification/Debugging Files カテゴリ内より “Signal Tap Logic Analyzer File” を選択します。

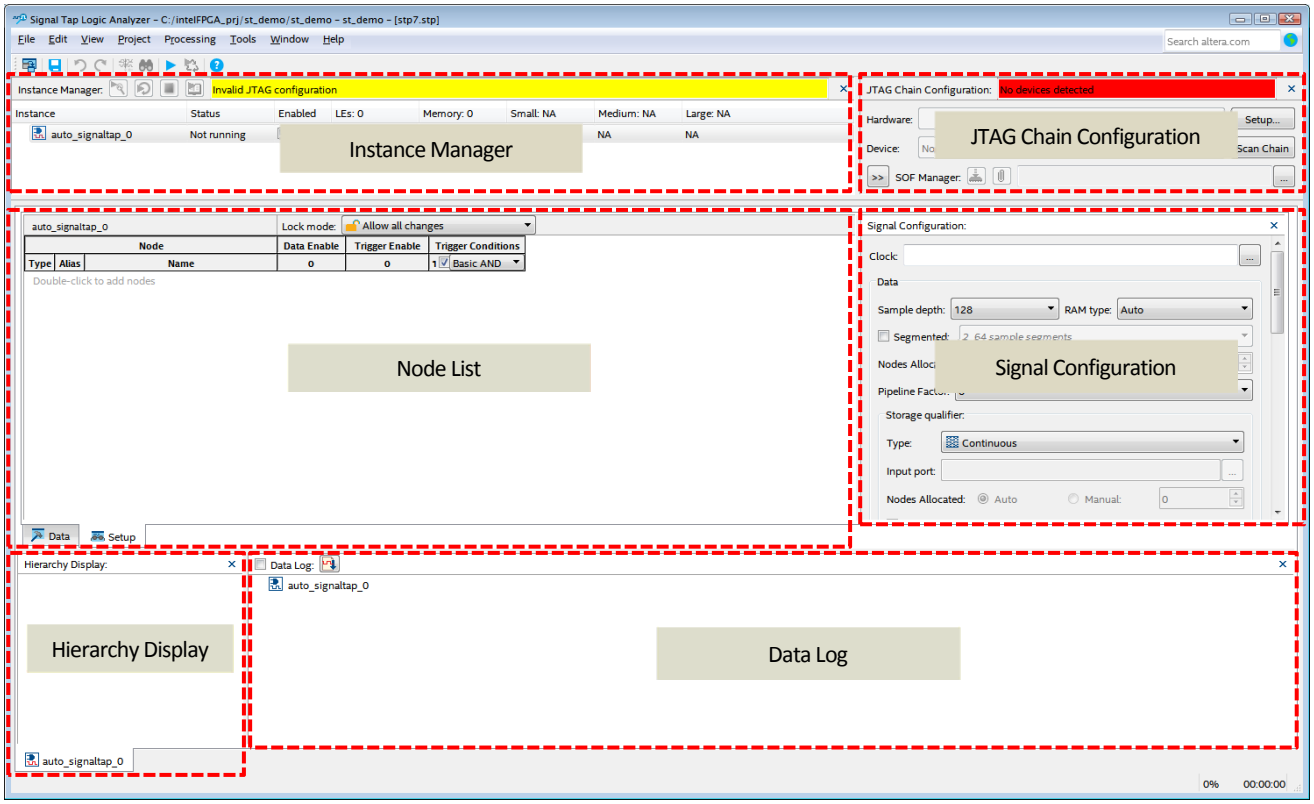
プロジェクトに既存登録されている STP ファイルがない場合には、Tools メニュー ➤ Signal Tap Logic Analyzer からも新規 STP ファイルが起動します。



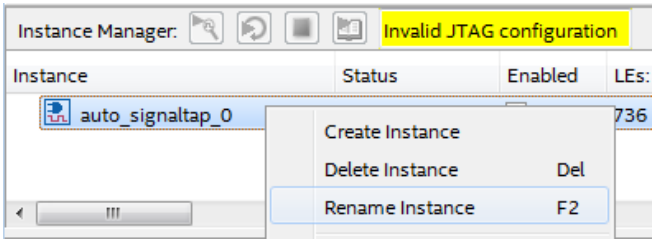


プロジェクトに STP ファイルが既存登録されている場合は、その STP ファイルが起動します。

6 つのペインで構成された STP ファイルが表示されます。

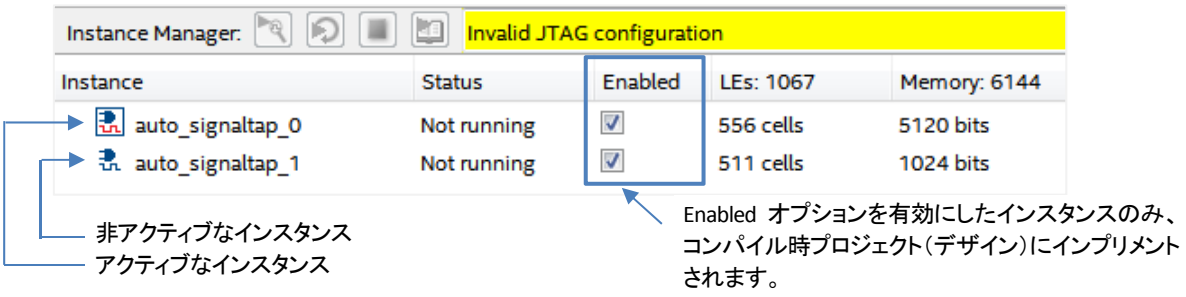


Instance Manager ペインに auto_signaltap_0 という名称の Signal Tap IP のインスタンスが自動的に作成されます。インスタンス名を変更する際は、インスタンスを選択して右クリック ➤ Rename Instance より行います。



Signal Tap IP コアは、1 つの FPGA に複数インスタンスすることが可能です。Signal Tap IP コアを追加する場合は、Instance Manager ペイン内で右クリックし、Create Instance を選択してください。複数インスタンスすると、Signal Tap IP コアを構成するためのロジック・エレメントやメモリの消費が増加するため、空きリソース容量の範囲で作成する必要があります。

各インスタンスは、別々のクロックを用いられることや同一クロック・ドメインに対して検証する信号ごとにインスタンスを割り当てることができるほか、個々にデバッグが実行できるので、ユーザのニーズに合ったデバッグ方法が実現できます。操作がアクティブなインスタンスは、アイコンの波形が赤く表示され、ボックスで囲われます。各インスタンスの操作切り替えは、Hierarchy Display ペインのタブにより指定します。

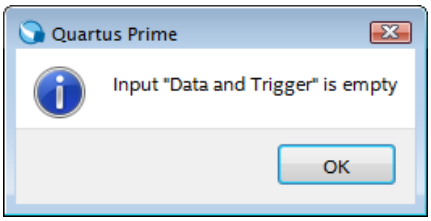


4-1-2. デザインに STP ファイルを追加

Signal Tap IP を駆動させるためのクロックやサンプリング信号の登録、トリガ条件の設定などがまだ行われていませんが、先に STP ファイルに名前を付け保存し、プロジェクトに登録しておきます。

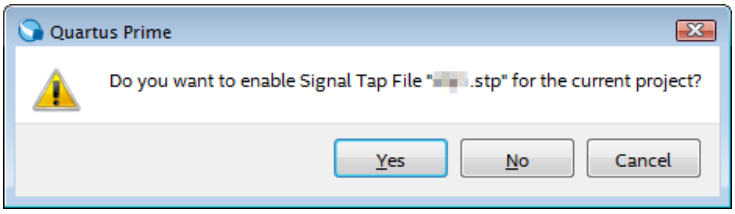
ユーザ・デザインと Signal Tap IP のポート接続は、STP ファイルを作成しプロジェクトに登録することで Quartus Prime が自動的にユーザ・デザインと Signal Tap IP を接続してくれます。

① 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、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。

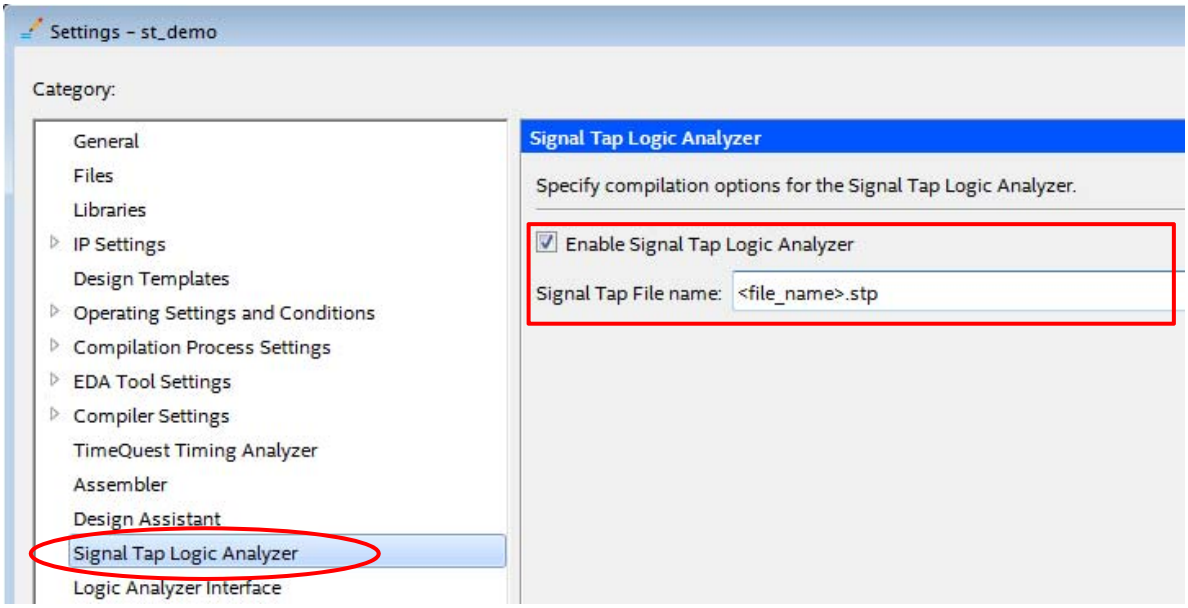


← STP ファイルに何も入力していない場合には *Input “Data and Trigger” is empty* とメッセージ（左図）が表示されます。

② *Do you want to enable Signal Tap File “<任意ファイル名>.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。



この操作により、Signal Tap Logic Analyzer オプション（Assignments メニュー ➤ Settings）を自動的に有効にし、ユーザ・デザインにインスタンスする STP ファイルを指定したことになります



4-2. Signal Tap の設定

STP ファイルに、Signal Tap IP 用クロックや観測する信号、およびトリガとして用いる信号などを登録します。

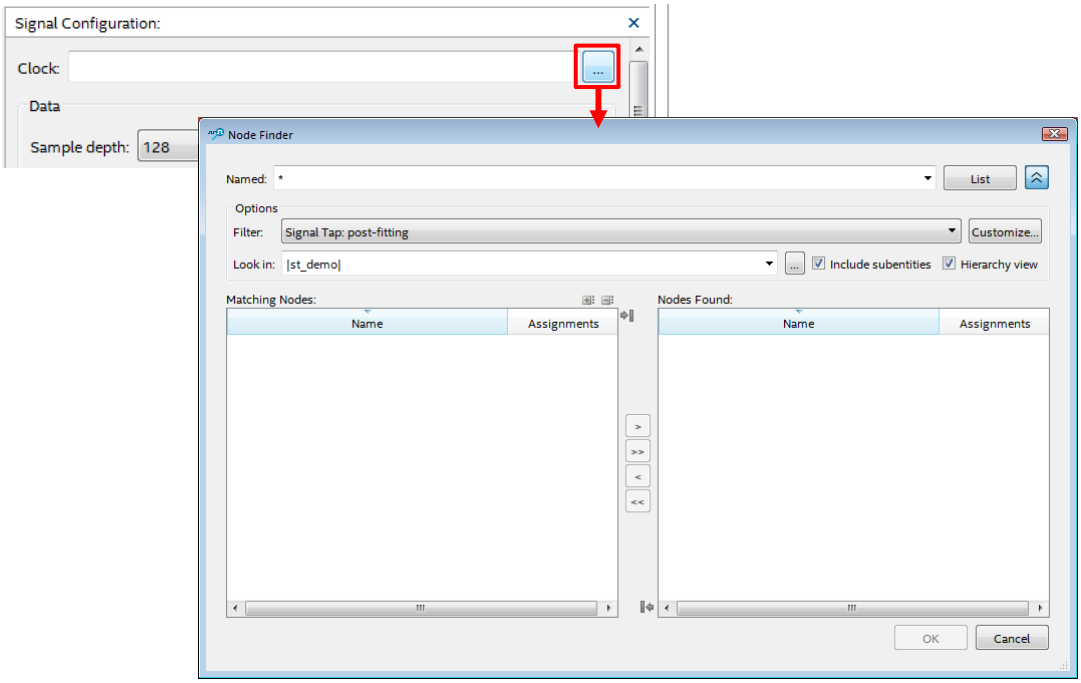
4-2-1. クロックの設定

Signal Tap IP のクロックを指定します。

Signal Tap IP はクロックの立ち上がりでデータをサンプリングします。立ち下りクロックでのサンプリングはサポートされていません。このクロックは、デザイン内のどの信号でも選択することができますが、最良の結果を得るため、サンプリングする信号に同期した、かつグローバル化されたクロックの使用を推奨しています。また、サンプリングするデータに対し 2 倍以上の周波数のクロックを割り当てると良いでしょう。

① STP ファイルの Setup タブをクリックします。

② Signal Configuration ペインの 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。



③ Node Finder ウィンドウ内 Filter 欄のプルダウン・リストから、Signal Tap IP 用クロックとして採用する信号のタイプ（以下参照）を選択し、List ボタンをクリックします。

- インクリメンタル・コンパイルを採用していない場合 ⇒ Signal Tap: pre-synthesis を選択
- インクリメンタル・コンパイルを採用している場合 ⇒ Signal Tap: post-fitting を選択

選択するクロックがユーザ・デザインにおいて深い階層下にある場合は、Look in 機能により階層を指定して信号を検索することが可能です。

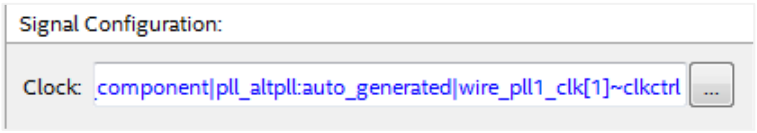
Look in 欄右のブラウズ・ボタンをクリックし、Select Hierarchy Level ダイアログ・ボックスから目的のエンティティを選択すると、指定した階層名が Look in 欄に表示されます。その後 Node Finder の List ボタンをクリックすると、目的の信号が見つけやすくなります。

Node Finder ウィンドウの Named 欄にはワイルドカードの使用が可能ですので、組み合わせて検索するとより検索がスムーズになります。

④ 検出された Matching Nodes リストの信号から、クロックにアサインしたい信号をダブルクリックで指定し、Nodes Found リストに選出します。

⑤ OK ボタンをクリックし、Signal Tap IP のクロックを登録します。

エントリされた信号の文字色は、Post-fitting 信号は青、Pre-synthesis 信号は黒で表示されます。



4-2-2. 信号の登録

観測する内部信号を選択します。選択した信号はトリガ条件にも使用できます。ただし Signal Tap IP 用のクロックに指定した信号は観測できません。

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。

② Node Finder ウィンドウ内 Filter 欄のプルダウン・リストにおいて、以下のいずれかを指定します。

- インクリメンタル・コンパイルを採用していない場合 ⇒ Signal Tap: pre-synthesis を選択
- インクリメンタル・コンパイルを採用している場合 ⇒ Signal Tap: post-fitting を選択

選択するクロックがユーザ・デザインにおいて深い階層下にある場合は、Look in 機能により階層を指定して信号を検索することが可能です。

Look in 欄右のブラウズ・ボタンをクリックし、Select Hierarchy Level ダイアログ・ボックスから目的のエンティティを選択すると、指定した階層名が Look in 欄に表示されます。その後 Node Finder の List ボタンをクリックすると、目的の信号が見つけやすくなります。

Node Finder ウィンドウの Named 欄にはワイルドカードの使用が可能ですので、組み合わせて検索するとより検索がスムーズになります。

Note: STP ファイルに観測したい、あるいはトリガとして用いたい内部信号 (Signal Tap: post-fitting) を選出する際、Node Finder を使用せずに Technology Map Viewer (Post-Fitting) を活用することも可能です。ユーザ・デザインの内部信号を視覚的に検索し、STP ファイルにインポートさせます。詳細は、下記 TIPS をご参考ください。

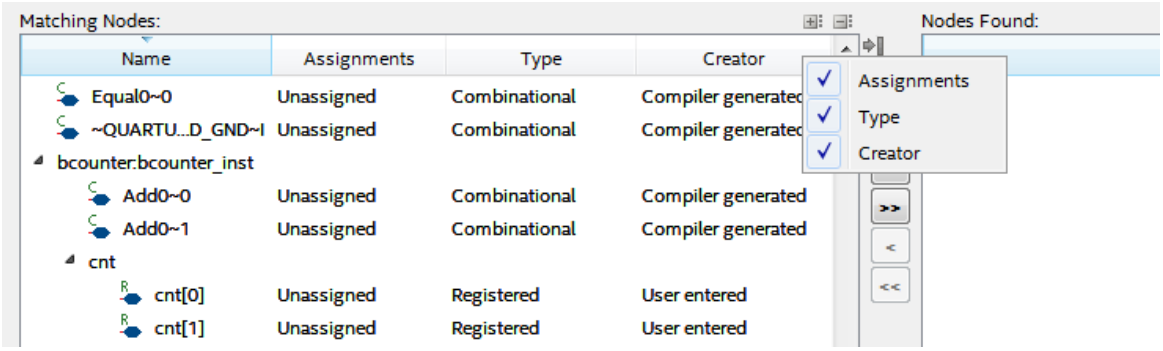
【TIPS】SignalTap® II でキャプチャしたい内部信号を簡単に STP ファイルに登録する方法

③ 検出された Matching Nodes リストの信号から、選出したい信号名を Nodes Found リストに登録します。キーボードの Shift または Ctrl キーを使用して、一度に複数の信号を選択することができます。

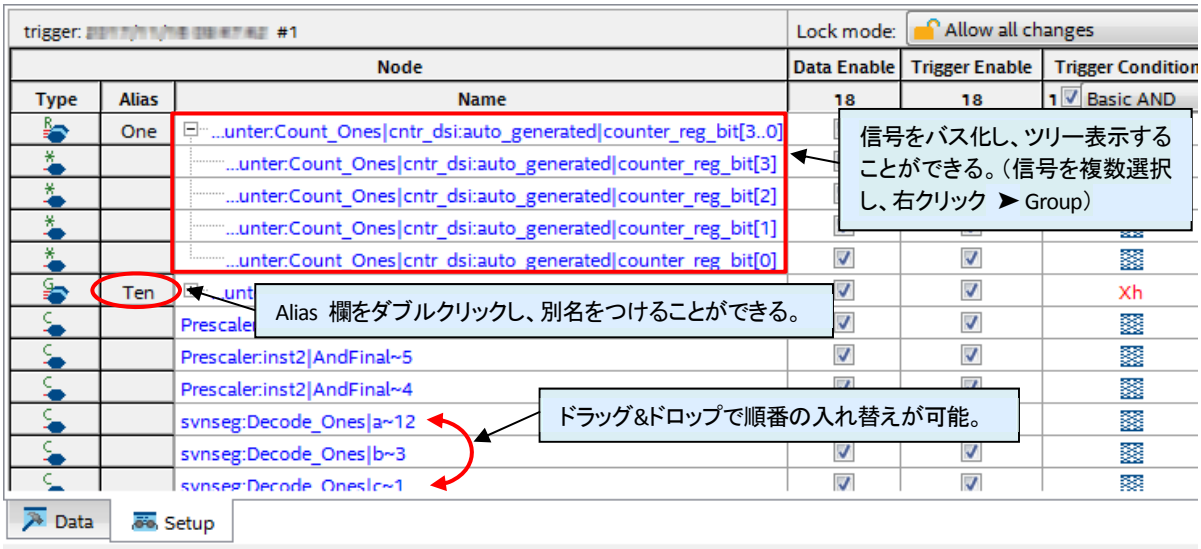
Node Finder にリストアップされる信号名の左側に表示されるピンスタブ（マーク）は、信号の種類を表しています。

ピンスタブ	種類
	入力ピン信号
	出力ピン信号
	組み合わせ信号
	レジスタ出力信号
	バス信号（ピンスタブが重なっている）

また、カラムを右クリックで選択し項目を追加することができます。Type では信号の種類、Creator ではユーザ作成の信号名か Quartus Prime のコンパイラが生成した信号名かを確認することができます。



④ Insert ボタンをクリックし、信号を登録します。Node List エリアに表示された信号の文字色は、Post-fitting 信号は青色、Pre-synthesis 信号は黒色を示しています。コンパイル後に存在しない信号は赤色で表示され、検証に使用できません、STP ファイルから削除してください。



ユーザ・デザインにおいて多くの信号は、Quartus Prime のコンパイラにより論理合成また配置配線で最適化が図られます。これにより、RTL で記述した信号名とは異なってしまい、ユーザは Signal Tap で検証したい信号を検索することが困難になるかもしれません。インクリメンタル・コンパイルを使用した場合も同様です。その場合、先にも記載した Technology Map Viewer の活用が有効的です。

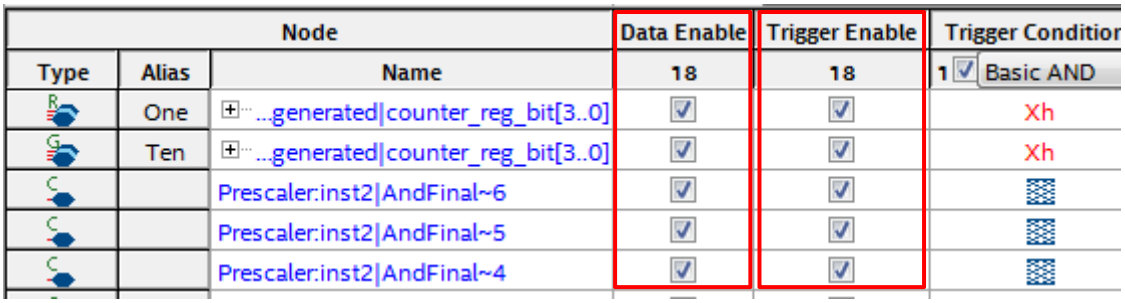
その他の対策として、論理合成および配置配線時に信号を保持させるアトリビュートを採用する方法があります。

- Keep : 指定した組み合わせ信号が削除されないようにします。
- Preserve : 指定したレジスタが削除されないようにします。

このアトリビュートは、ユーザ・デザインに対してオプションとして採用する方法と HDL 記述に直接宣言する方法のいずれかで適用させます。ただしこの方法はコンパイラによる最適化が行われないため、リソースの使用率が増加したり、タイミング性能が低下する場合があります。これらのアトリビュートの使用については、メーカのドキュメント「Intel Quartus Prime Integrated Synthesis」を参照してください。

STP ファイルに登録した信号には Enable オプションがあり、各々使用方法を指定することが可能です。いずれのオプションも、デフォルトは有効（ON）です。

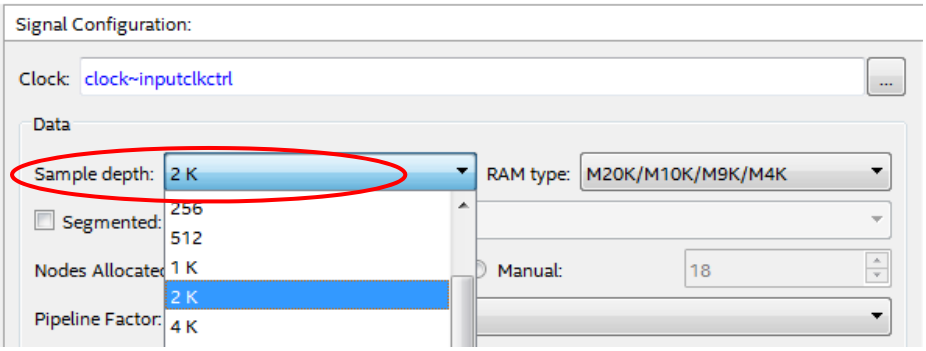
- Data Enable ： このオプションをディセーブル（OFF）にすると、指定した信号のデータを非表示にします。例えば、信号はトリガに用いるのみで、その信号自体のデータは表示しなくて良い場合に設定します。これにより、Signal Tap IP で使用される FPGA 内部メモリの使用率を削減することができます。
- Trigger Enable: このオプションをディセーブル（OFF）にすると、トリガ機能を無効にします。キャプチャしたデータのみを表示し、その信号をトリガの一部として使用しない場合に設定します。



4-2-3. サンプル容量の指定

サンプル容量とは、各信号に対してキャプチャおよび格納されるサンプルの数です。Signal Configuration ペイン内 Data エリアの Sample depth において、プルダウン・リストから格納する希望サンプル数を選択します。サンプル容量の範囲は 0～128K です。

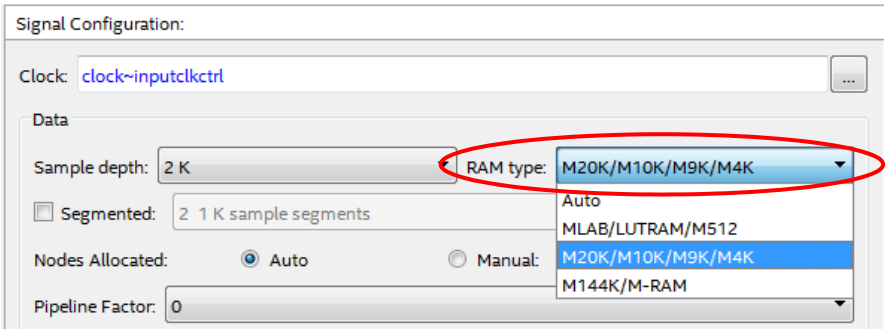
FPGA に搭載されたメモリ・リソースの空き容量が、選択したサンプル・バッファ・サイズではデザインをコンパイルできない場合があります。そのときはサンプル容量を少なくしてリソースの使用率を低減してください。



4-2-4. RAM タイプの指定

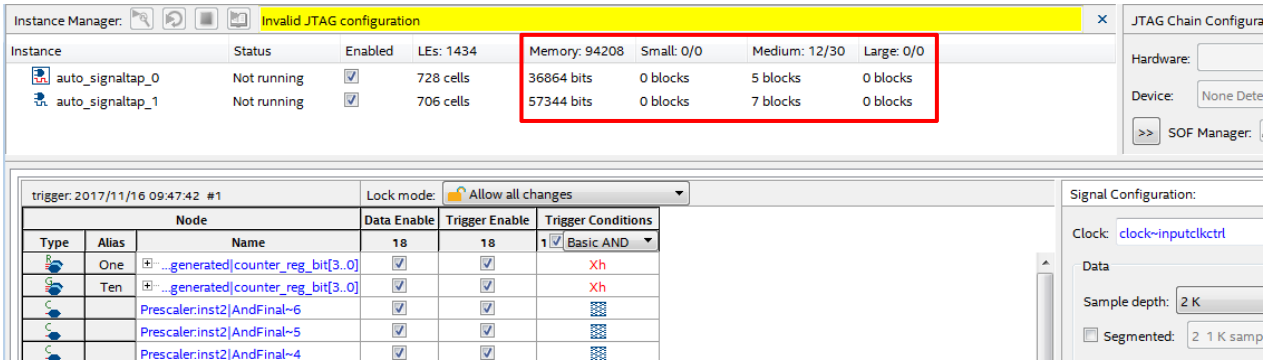
観測データを格納する内部メモリ・ブロックの種類を指定します。Signal Configuration ペイン内 Data エリアの RAM type において、プルダウン・リストから選択します。（ターゲット・デバイスのアーキテクチャにより、RAM タイプが指定できないファミリもあります。）

もし、ユーザ・デザインに大容量メモリ・アプリケーションがある場合、Signal Tap で使用する RAM タイプを小規模あるいは中規模メモリ・ブロックに割り当てることで、大規模なメモリ・ブロックをユーザ・デザインのために有効活用することができます。



一つのインスタンスで RAM タイプを混在させることはできません。インスタンスを複数作成した場合は、インスタンスごとに RAM タイプの指定が可能です。インスタンスを複数作成する方法は、本資料 4-1-1. STP 新規 STP ファイルを起動 をご覧ください。

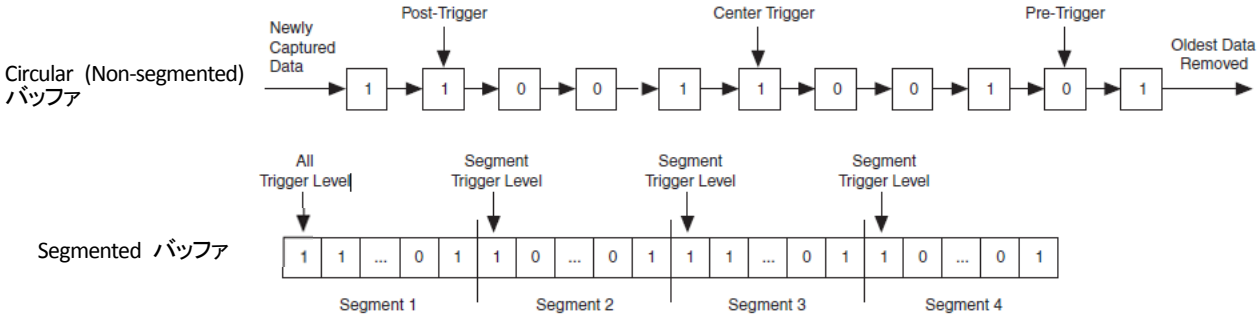
RAM タイプを指定すると、Instance Manager ペインに使用されるメモリの容量とブロック数が自動計算され表示されます。



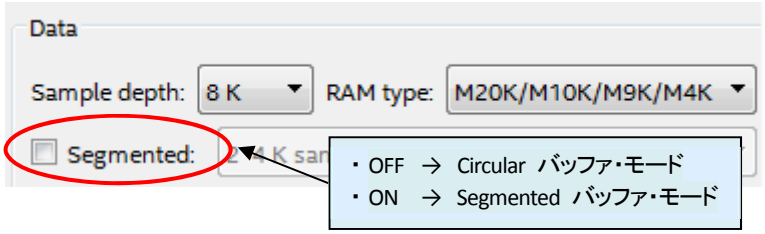
4-2-5. バッファ・モードの選択

キャプチャしたデータ・バッファの収集方法を選択でき、データ収集に必要なメモリ容量を潜在的に低減することができます。収集方法は以下の 2 つです。

Circular (Non-segmented) バッファ	デフォルトのバッファ・タイプ。全てのメモリ領域を一つの FIFO とみなし、ロジック・アナライザが定義したトリガが発生するまで、連続的にバッファを満たします。トリガ発生後もバッファがいっぱいになるまでデータのキャプチャを継続します。
Segmented バッファ	メモリ領域をセグメントに分割し、個々のセグメントは単体の Circular バッファのように機能します。周期的に発生するトリガをキャプチャするのに適した方法です。



バッファ・モードは、Signal Configuration ペイン内 Data エリアの Segmented オプションで選択します。

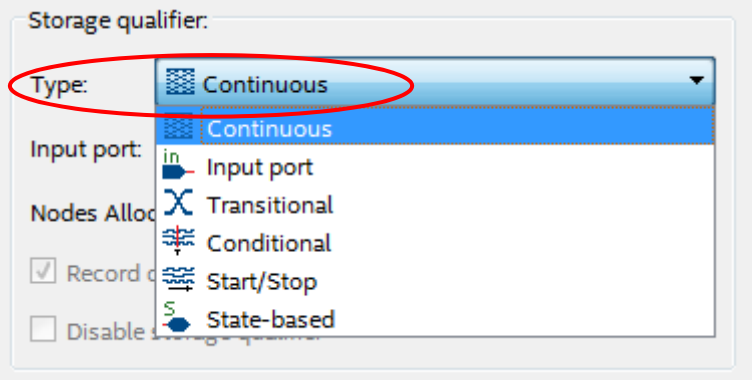


この資料では、Circular バッファ・モードに関して説明します。Segmented バッファ・モードの詳細は、下記ドキュメントをご覧ください。

「Design Debugging with the Signal Tap Logic Analyzer」(Quartus Prime Standard Edition)

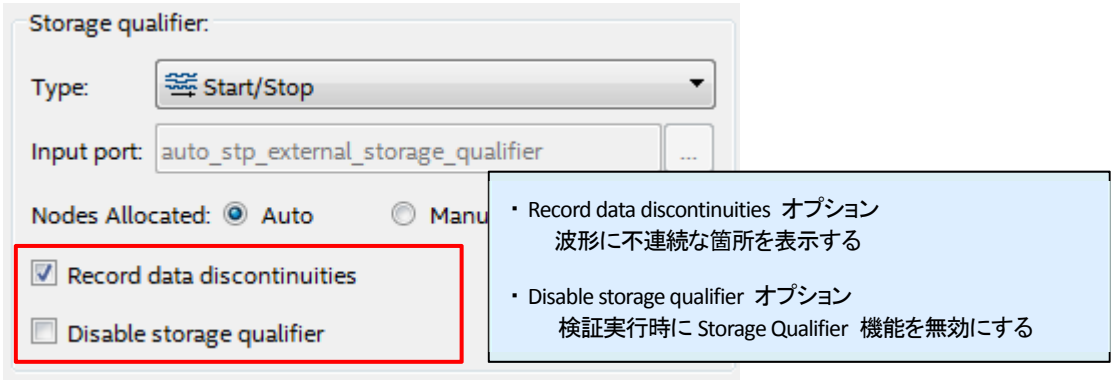
「Design Debugging with the Signal Tap Logic Analyzer」(Quartus Prime Pro Edition)

Circular バッファ・モードでは、Storage Qualifier 機能を選択して、限られたバッファ・スペースをより有効的に活用できます。Signal Configuration ペイン内 Storage qualifier エリアで指定できるオプションの種類は以下のとおりです。



Storage Qualifier の種類	
Continuous	全サンプルをストア。
Input port	デザイン内の指定した信号がクロック・エッジに伴い High のときにサンプルをストア。(バッファのライトイネーブルとして使用)
Transitional	選択された信号の遷移時にサンプルをストア。
Conditional	選択した信号のブール式が真のときサンプルをストア。
Start/Stop	スタートに指定した信号が真のときからストップに指定した信号が真になるまでサンプルをストア。
State-based	ステートマシンのフローを制御し、ステート状態が真のときにサンプルをストア。(詳細は Trigger Condition Flow Control をご覧ください。)

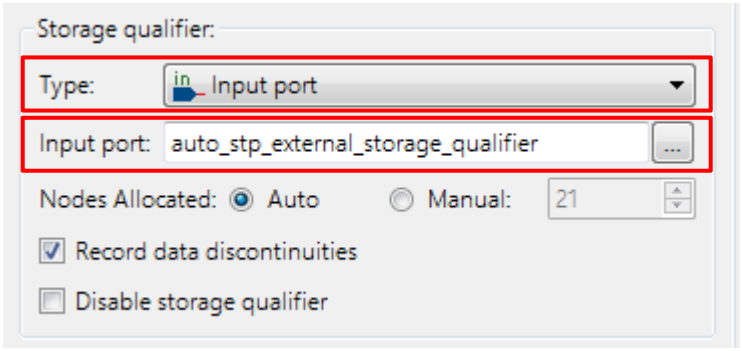
また、Storage qualifier エリアに設けられたオプションを併用することも可能です。



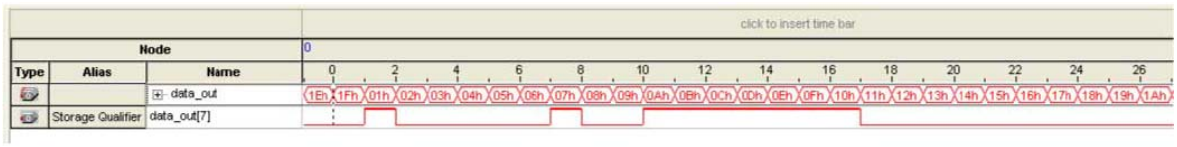
以下に、各 Storage Qualifier Type の波形サンプル（State-based を除く）を表示します。State-based モードについては、Trigger Condition Flow Control をご覧ください。

◆ Input Port モード

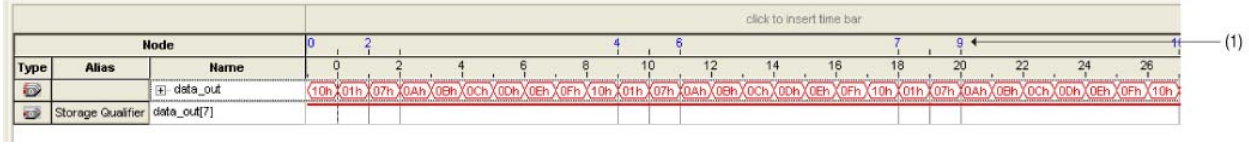
指定した信号の論理値が High 時のデータをストアします。



・ Continuous Mode

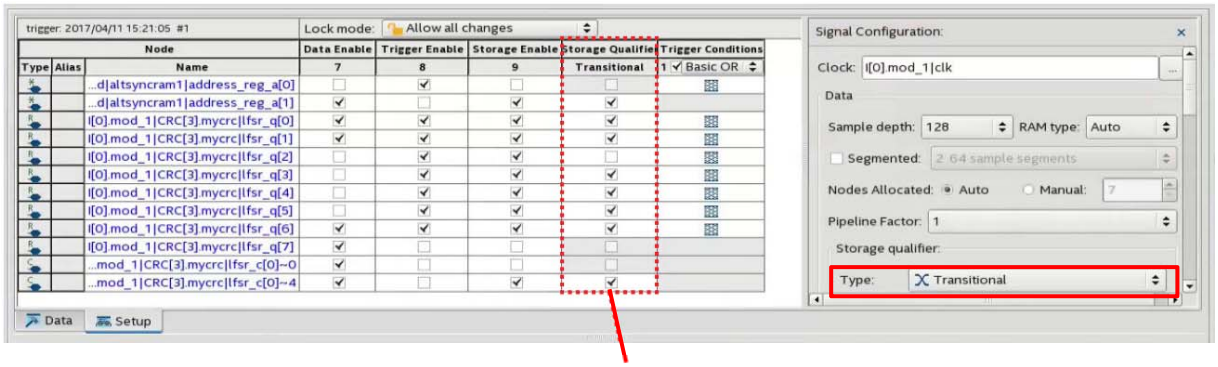


・ Input Port Storage Qualifier



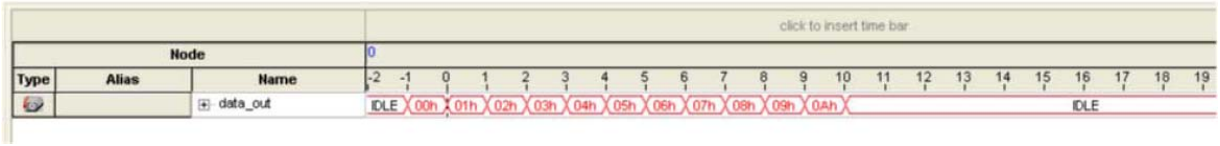
◆ Transitional モード

モニタ対象に指定した信号のいずれかが変化した場合にデータをストアします。

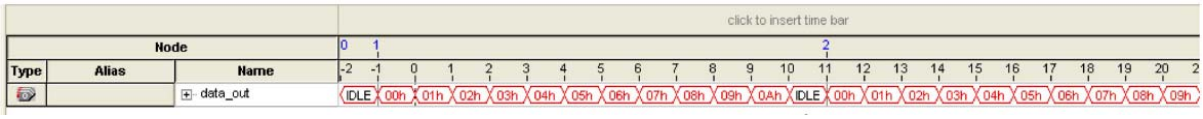


モニタする信号を選択

- Continuous Mode

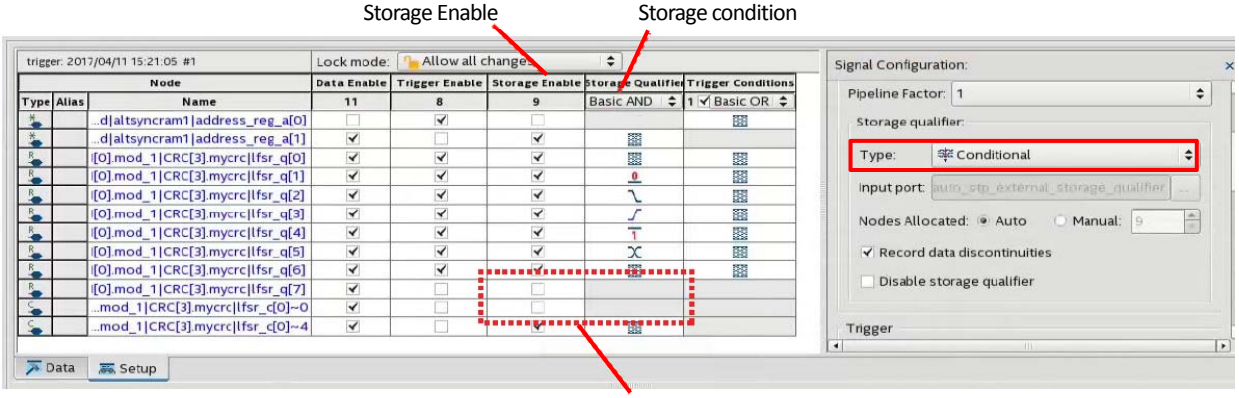


- Transitional Mode



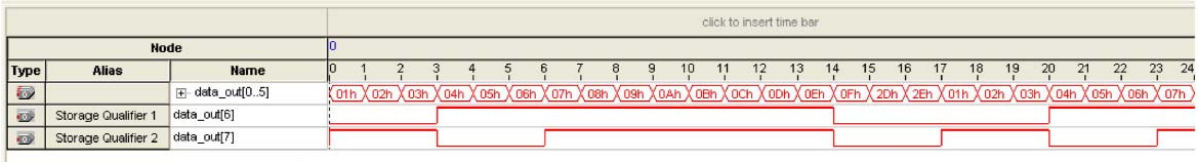
◆ Conditional モード

特定の条件成立時にデータをストアします。



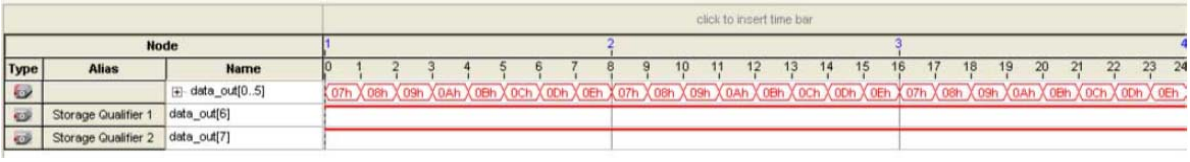
信号の Storage Enable が有効になっていない場合、信号は Storage Qualifier 条件の一部になることはできません。

- Continuous Mode



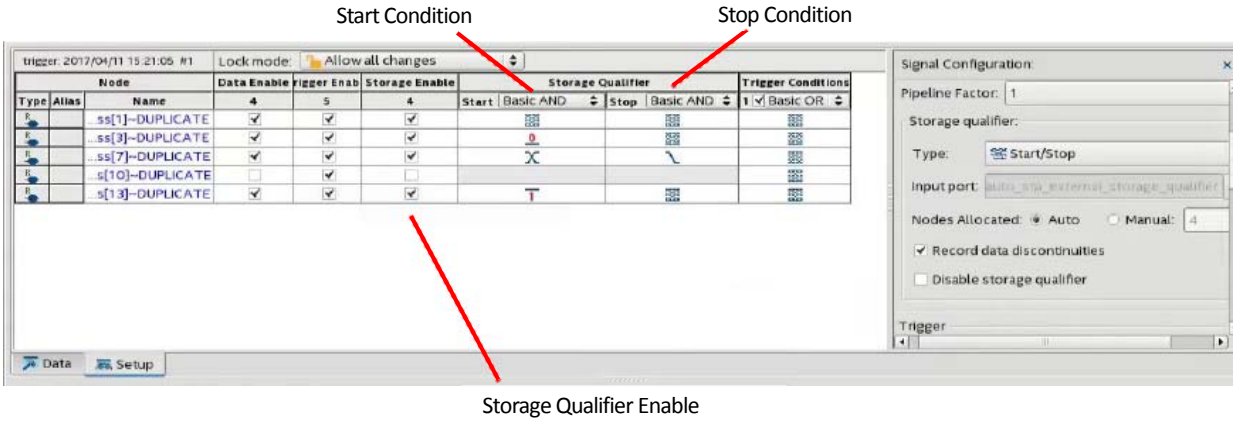
- Conditional Mode

※ Storage Qualifier 条件は、data_out[6] AND data_out[7] を評価するように設定した場合

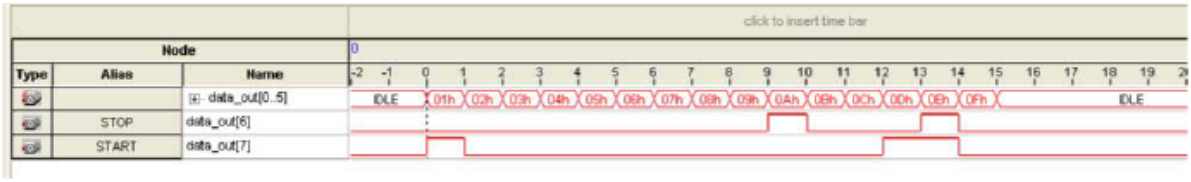


◆ Start/Stop モード

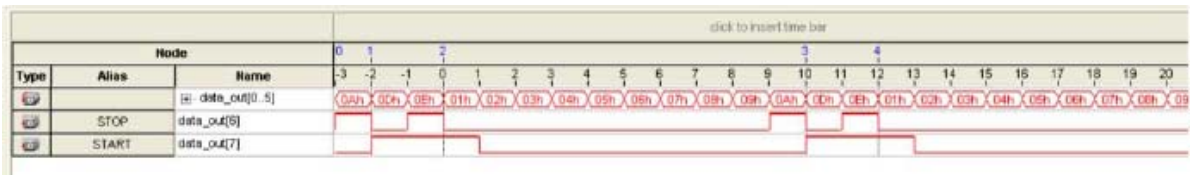
Start/Stop 2 つの条件を設定し、成立時にデータをストアします。なお、Stop 条件は Start 条件よりも優先されます。



- Continuous Mode



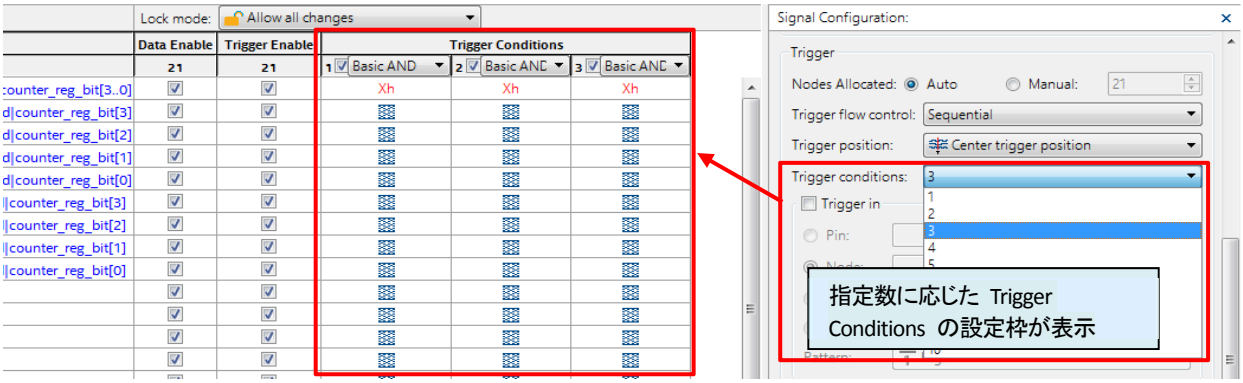
- Start/Stop Storage Qualifier:



4-3. トリガの定義

モニタする信号のトリガ条件を指定します。

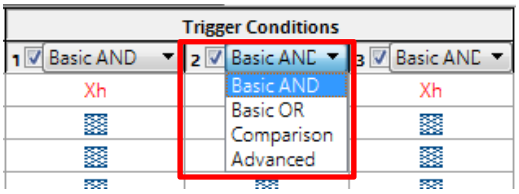
Signal Configuration ペインの Trigger Conditions でトリガ条件を複数指定できます。最大 10 レベルのトリガ・コンディションを設定可能です。



4-3-1. トリガ・コンディションの設定

必要なデータをキャプチャするため、モニタする信号がそのデータを表示する条件(トリガ)を指定します。

Setup タブの Node List 内、Trigger Conditions 項目のプルダウン・リストにより、トリガ・コンディション・タイプを選択します。タイプは 4 つあります。



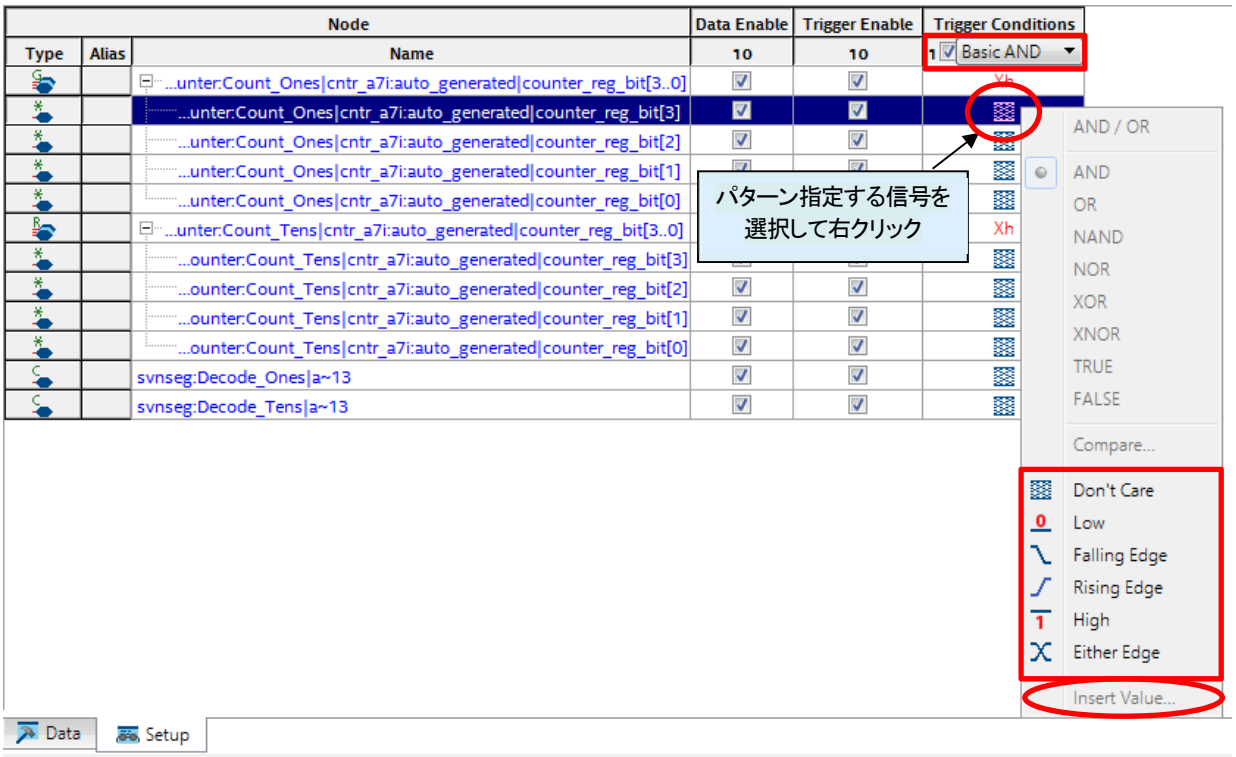
Basic AND / Basic OR	トリガに適用したい各信号にトリガ・パターンを指定
Comparison	バス信号上で簡単な比較条件を指定し、バスの複数のグループ化ビットを予想される整数値と比較
Advanced	専用エディタに GUI で演算子などを描き複雑なトリガ式(トリガ条件)を作成

◆ Basic AND / Basic OR トリガ

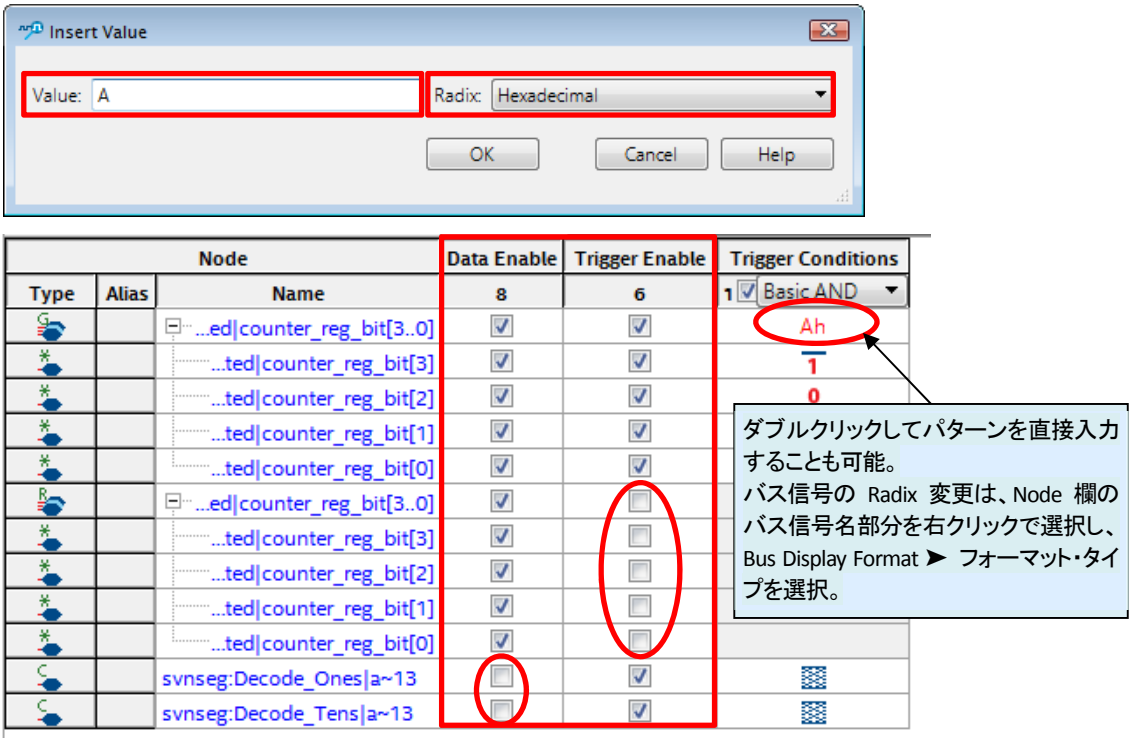
最もシンプルな種類のトリガで、追加した各信号のトリガ・パターンを設定する必要があります。

トリガ・パターンを設定するには、Trigger Levels カラムで右クリックし、希望のパターンをクリックします。トリガ・パターンは以下のいずれかに設定できます。

- Don't Care ・・・ 未定義(デフォルト)
- Low
- High
- Falling Edge ・・・ 立ち下りエッジ
- Rising Edge ・・・ 立ち上がりエッジ
- Either Edge ・・・ 立ち上がり または 立ち下りエッジ



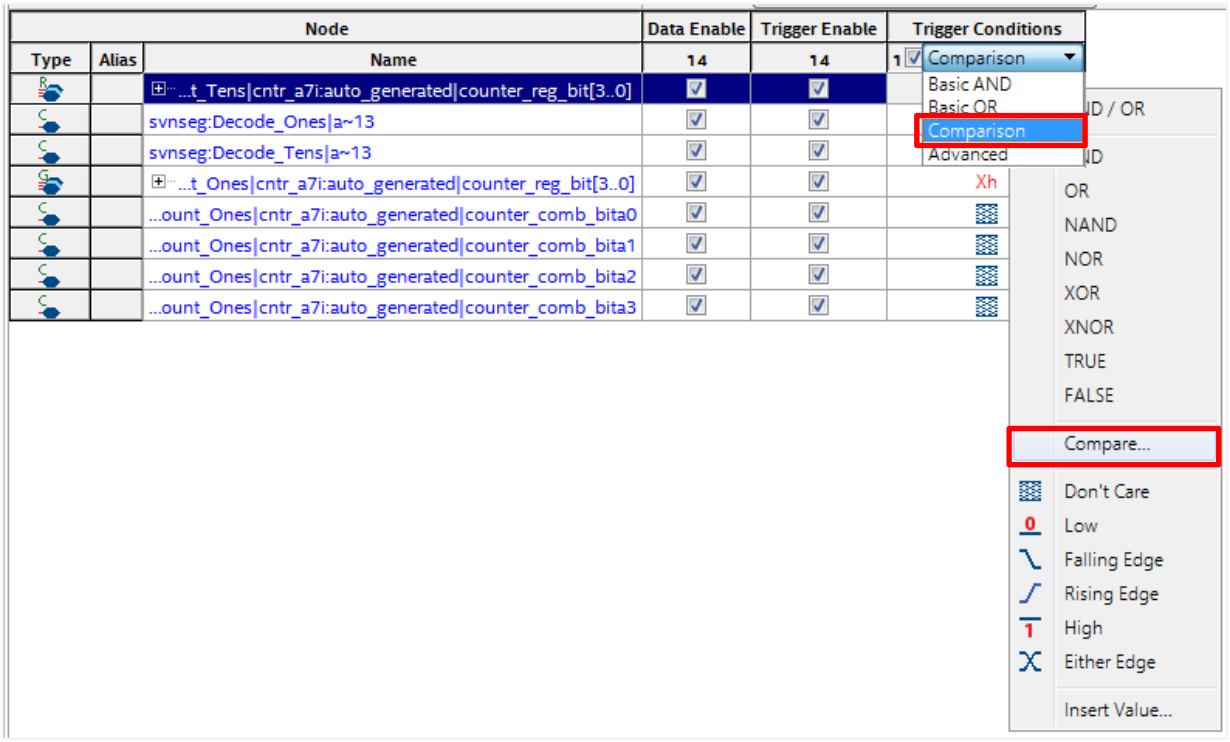
バス信号の場合は、グループ表示されたバス信号の Trigger Levels カラムを右クリック選択し、Insert Value から値をまとめて入力することが可能です。



選出した信号をトリガとして使用せずにデータの観測のみの場合には、Trigger Enable オプションを OFF します。同様に、選出した信号をトリガとしてのみ使用し、データの観測は必要がない場合には、Data Enable オプションは OFF にします。これにより、Signal Tap IP を構成するために必要なリソースを節約することが可能です。

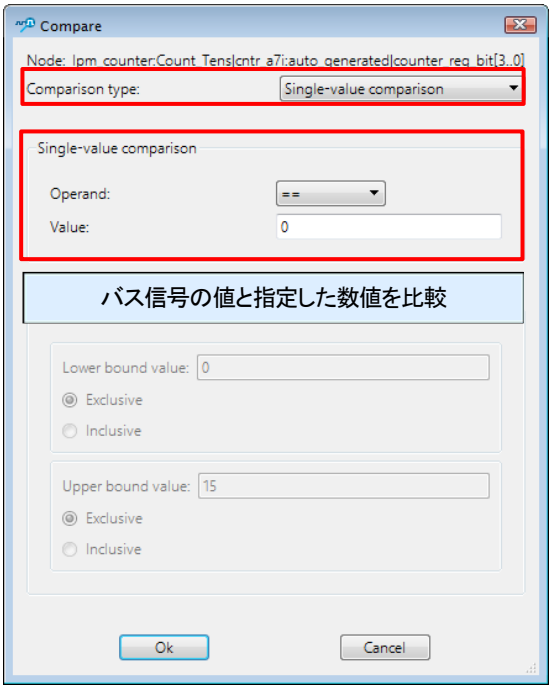
◆ Comparison トリガ

バス信号上で簡単な比較条件を指定することによって、バスの複数のグループ化ビットを予想される整数値と比較できます。Comparison トリガは Basic OR トリガに含まれるすべてのトリガ条件を保持します。

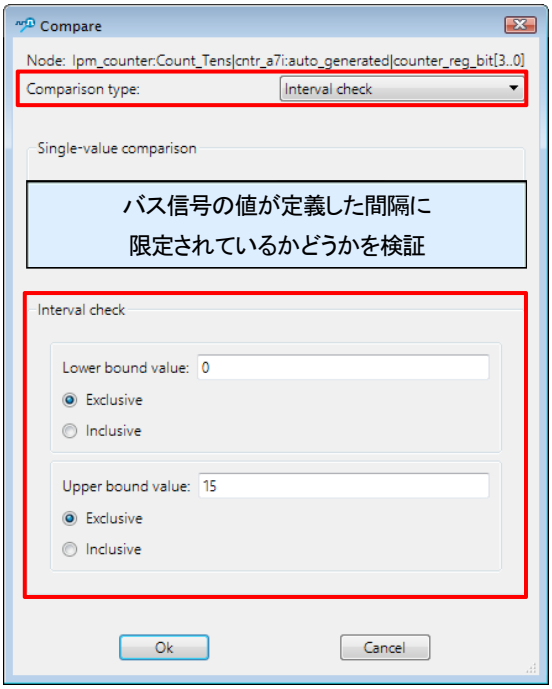


Comparison トリガは以下 2 つの比較タイプをサポートしています。

Single-value comparison

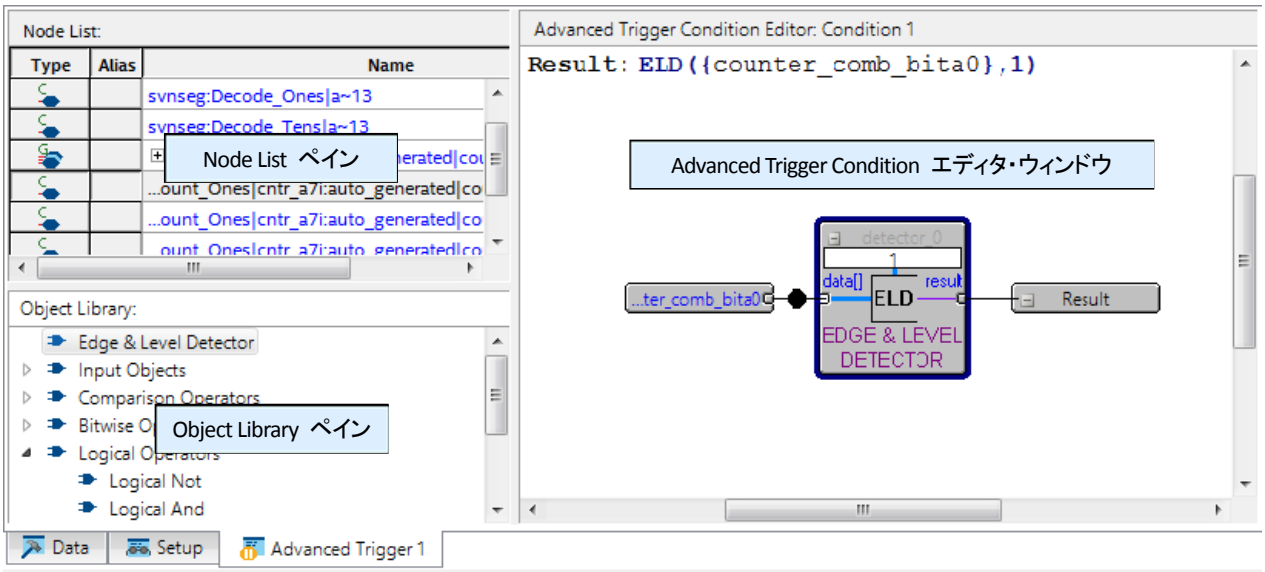


Interval check



◆ Advanced トリガ

トリガ・コンディション・タイプを Advanced に切り替えると、Advanced Trigger タブが追加され専用エディタが起動します。このエディタに Object や信号を用いて演算式を作成し、その演算結果がトリガ条件として使用されます。



Advanced トリガの詳細は、Advanced Trigger Conditions をご覧ください。

4-3-2. トリガ・フロー・コントロールの設定

トリガ・フロー・コントロールを設定し、作成したトリガ条件を正確に制御することができます。

Signal Configuration ペイン内 Trigger 項目の Trigger flow control プルダウン・リストにより選択します。タイプは以下の 2 つです。

- Sequential（デフォルト）
- State-Based

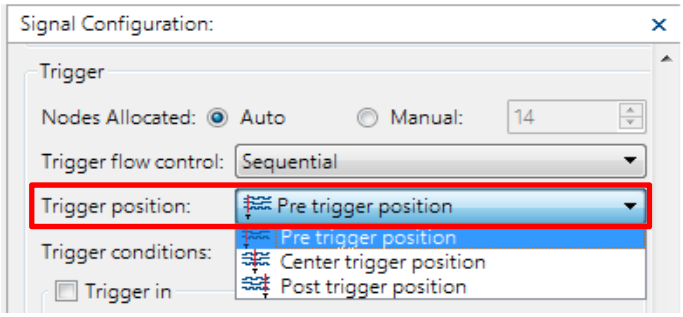
Sequential トリガは、取得バッファがキャプチャを終了する前に満たす必要のある最大トリガレベルを定義できます。

State-Based トリガについては、Trigger Condition Flow Control をご覧ください。

4-3-3. トリガ位置の指定

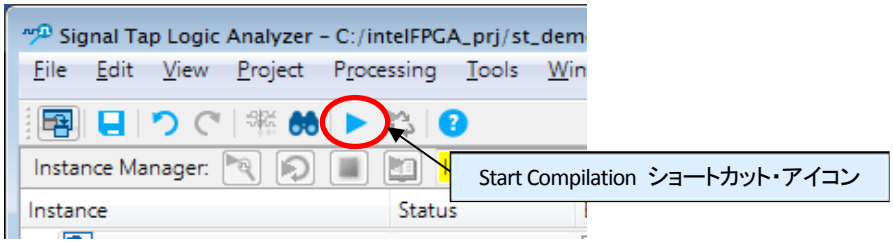
トリガ・イベントの前後に取得されるデータの量を指定します。

Pre trigger position	トリガの後に発生した信号の状態を保存します。(トリガ前 12%、トリガ後 88%)
Center trigger position	トリガ前とトリガ後のデータを半分ずつ保存します。(トリガ前 50%、トリガ後 50%)
Post trigger position	トリガの前に発生した信号の状態を保存します。(トリガ前 88%、トリガ後 12%)

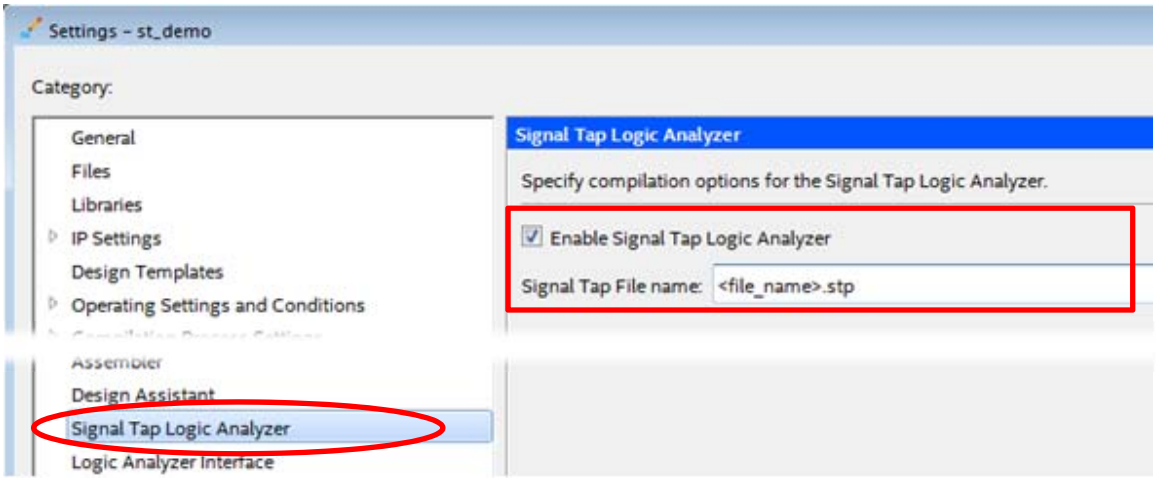


4-4. デザインのコンパイル

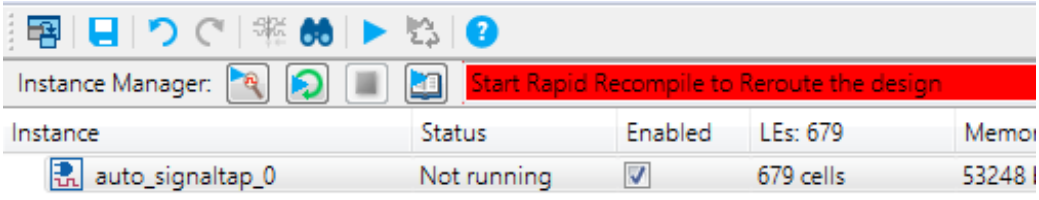
STP ファイルの Processing メニュー ➤ Start Compilation または Start Rapid Recompile(該当する FPGA のみ実行可能)を選択し、Signal Tap ロジック・アナライザ IP を組み込んだユーザ・デザインをコンパイルします。



Note: Signal Tap Logic Analyzer オプション(Assignments メニュー ➤ Settings)に設定された STP ファイルがユーザ・デザインにインスタンスされます。



Note: コンパイル完了後に STP ファイルの仕様変更を行った場合、それが Signal Tap IP に必要なリソースへ影響する変更であるときのみ再コンパイルが必要です。STP ファイルの Instance Manager ペインにもメッセージが表示されます。



4-5. FPGA プログラミング

デバッグを行う FPGA へ、コンパイルにより生成された SOF ファイルをダウンロードします。ダウンロードの操作は、STP ファイルで行います。

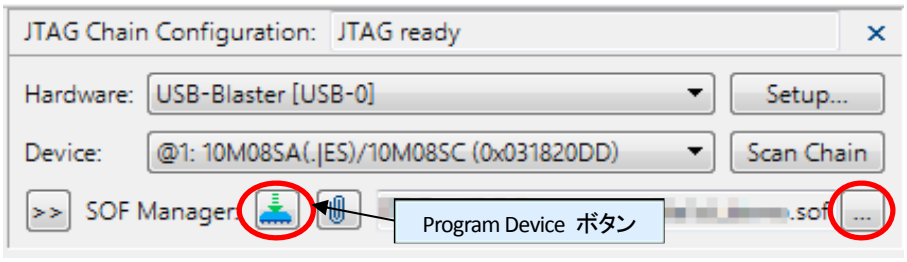
① パソコンとダウンロード・ケーブル、ダウンロード・ケーブルと FPGA の搭載されたボードを接続し、ボードの電源を投入します。

② STP ファイルの JTAG Chain Configuration ペインの Hardware 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。

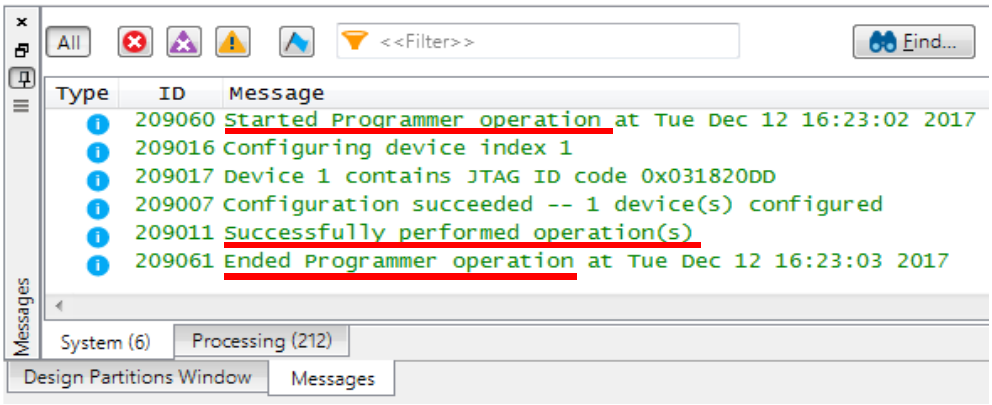
③ Scan Chain ボタンをクリック後、Device プルダウン・リストからターゲットの FPGA を選択します。

④ SOF Manager 右端のブラウズ・ボタンをクリックし、使用する STP ファイルを選択します。

⑤ SOF Manager 中央にある Program Device ボタンをクリックし、書き込みを実行します。



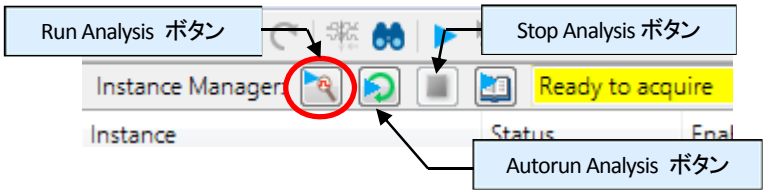
⑥ Messages ウィンドウに書き込みが完了したことを表すメッセージが確認できます。



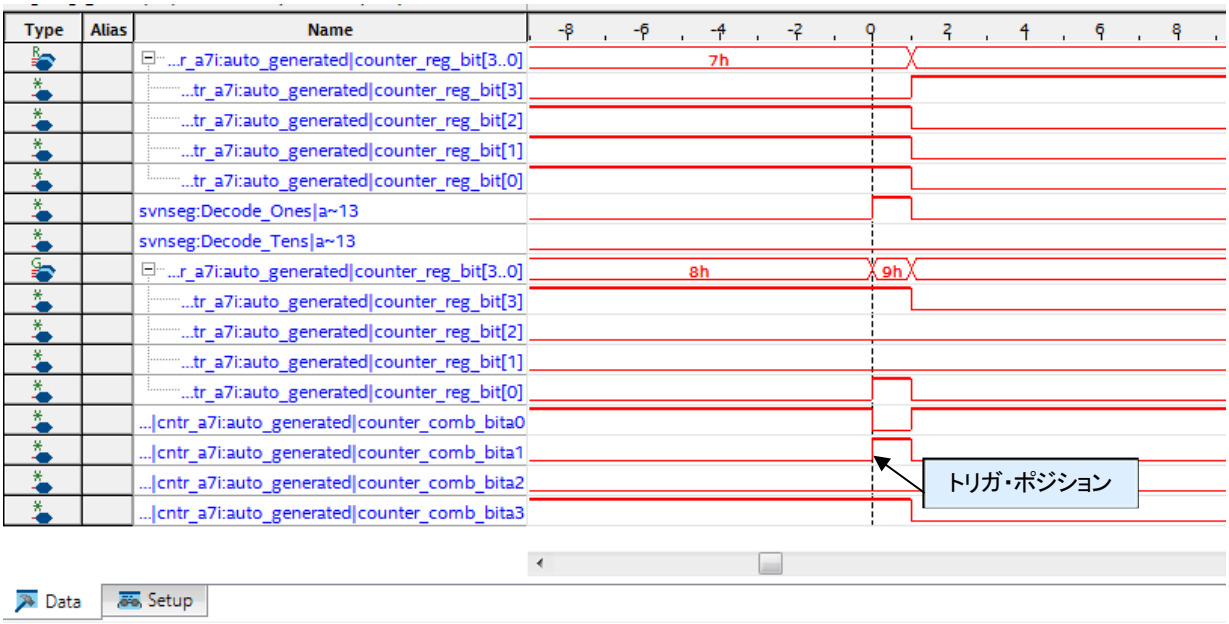
4-6. Signal Tap の実行

FPGA に SOFファイルが転送されると、すぐにSignal Tap IP を含んだユーザ・ロジックが動作し始め、STP に設定したトリガ条件に応じて、FPGA の内部メモリに指定した信号のデータが保存されています。

STP ファイルの Run Analysis ボタンをクリックし、メモリに格納されたデータを波形ウィンドウに表示させます。



Node List の Data タブ側に、検出された信号の波形が表示されます。



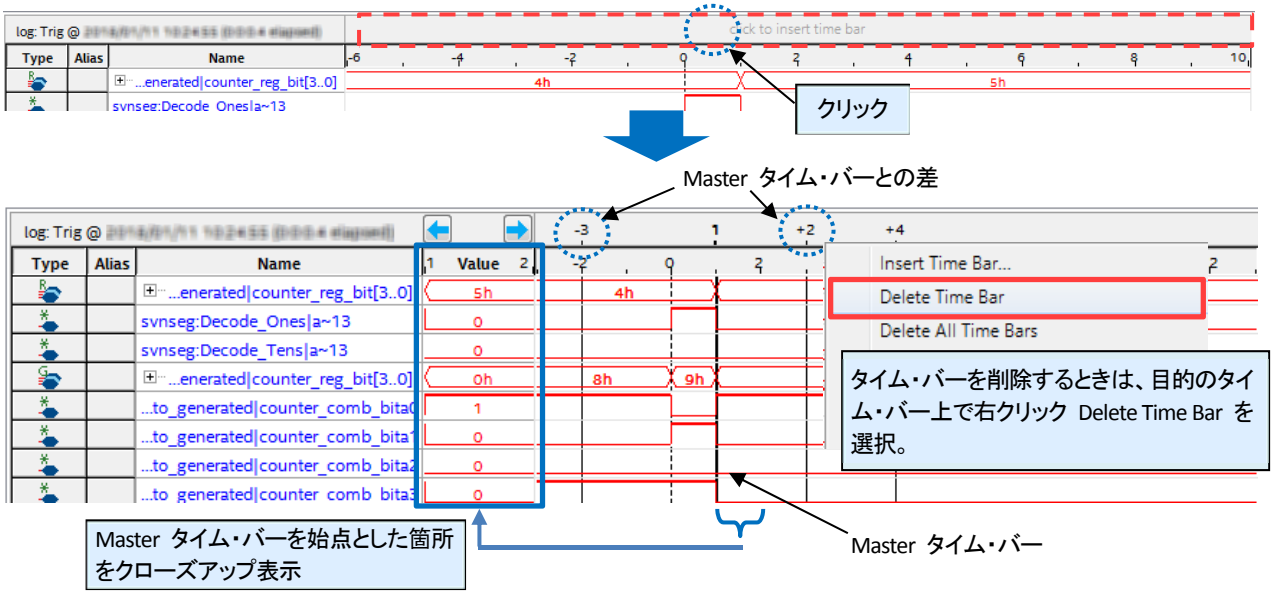
4-7. キャプチャされたデータの表示・解析

検出されたデータを解析する際に便利な機能がいくつかありますので、代表的なものを以下にご紹介します。

4-7-1. タイム・バー

波形表示ウィンドウに、タイム・バーを挿入することができます。

波形ウィンドウ上部（cick to insert time bar と表示された部分）の挿入したい場所をマウスでクリックします。



タイム・バーは ⬅ ➡ ボタンで左右へ移動可能です。マウスでタイム・バーをドラッグ&ドロップしても移動できます。

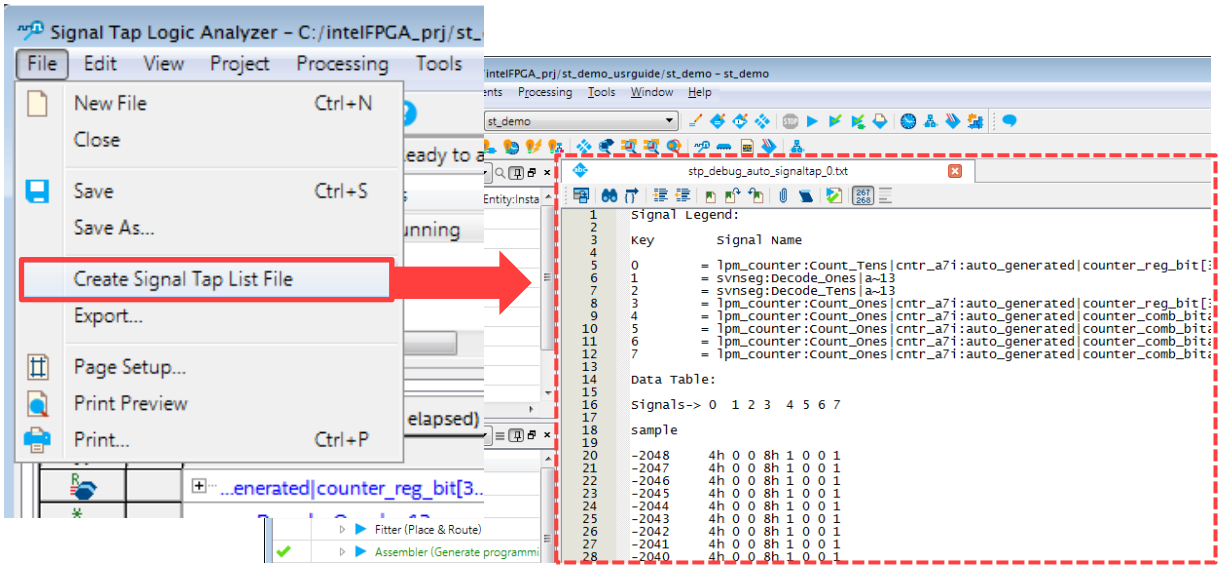
4-7-2. 検出データの変換

検出したデータをリスト・ファイルにすることができます。また、別のファイル・フォーマットに変換することも可能です。

リスト・ファイルの生成

① STP ファイル上の File メニュー ➤ Create Signal Tap List File を選択します。

② プロジェクト・フォルダに “*<STP ファイル名>_<Signal Tap インスタンス名>*.txt” が生成され、Quartus Prime のメイン画面にリスト・ファイルが表示されます。



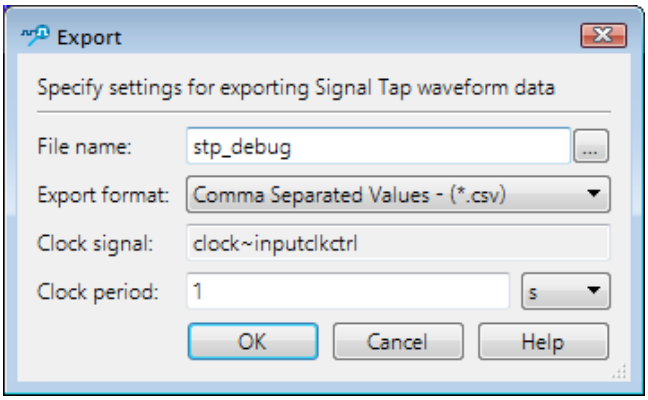
別のファイル・フォーマットに変換

① STP ファイル上の File メニュー ➤ Export を選択します。

② Export ウィンドウが表示されます。

- File name: 保存するファイル名を入力し、ブラウズ・ボタンで保存フォルダを指定します。
- Export format: プルダウン・リストから変換するフォーマットを選択します。
  - Comma Separated Values (.csv)
  - Table File (.tbl)
  - Value Change Dump (.vcd)
  - BMP Image File (.bmp)
  - JPEG Image File (.jpg)
- Clock signal: 使用したクロック信号名が表示されます。(自動)
- Clock period: クロック周期を入力します。

③ 指定したファイルが生成されます。



4-7-3. データ・ログの保存

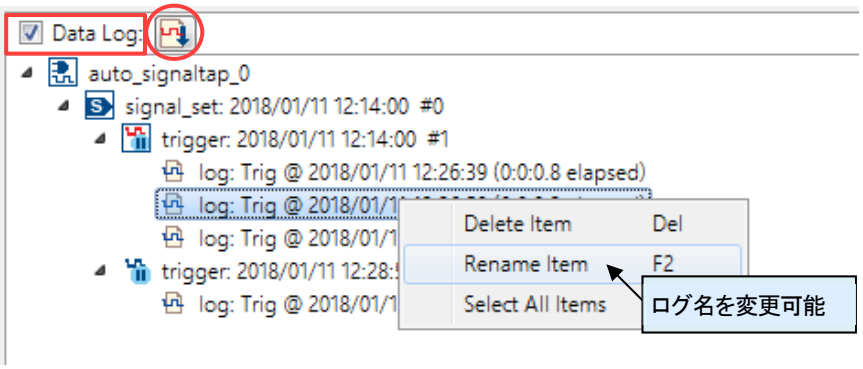
STP ファイルで使用したトリガ設定や、指定したトリガ条件で検出した波形データをログとして保存することができます。

① STP ファイルの Data log ペイン内にある Data Log オプションを有効にします。

② 右のボタン(Save to Data Log)をクリックすると、その時点で STP ファイルに表示されている各種情報が保存されます。

- signal_set : Signal Configuration ペインに登録されたログ
- trigger : トリガ設定のログ
- log : キャプチャされた波形のログ

これらはすべて、保存した日時が末尾に記録されます。



複数ログを保存した場合、表示させたいログをダブルクリックで選択してください。表示されているログのアイコンが枠で囲われます。

Data Log オプションを有効にした後に Signal Tap を実行(Run Analysis ボタンをクリック)した場合は、その度にキャプチャされた波形のログが追加記録されます。

改版履歴

Revision	年月	概要
1	2018 年 1 月	初版