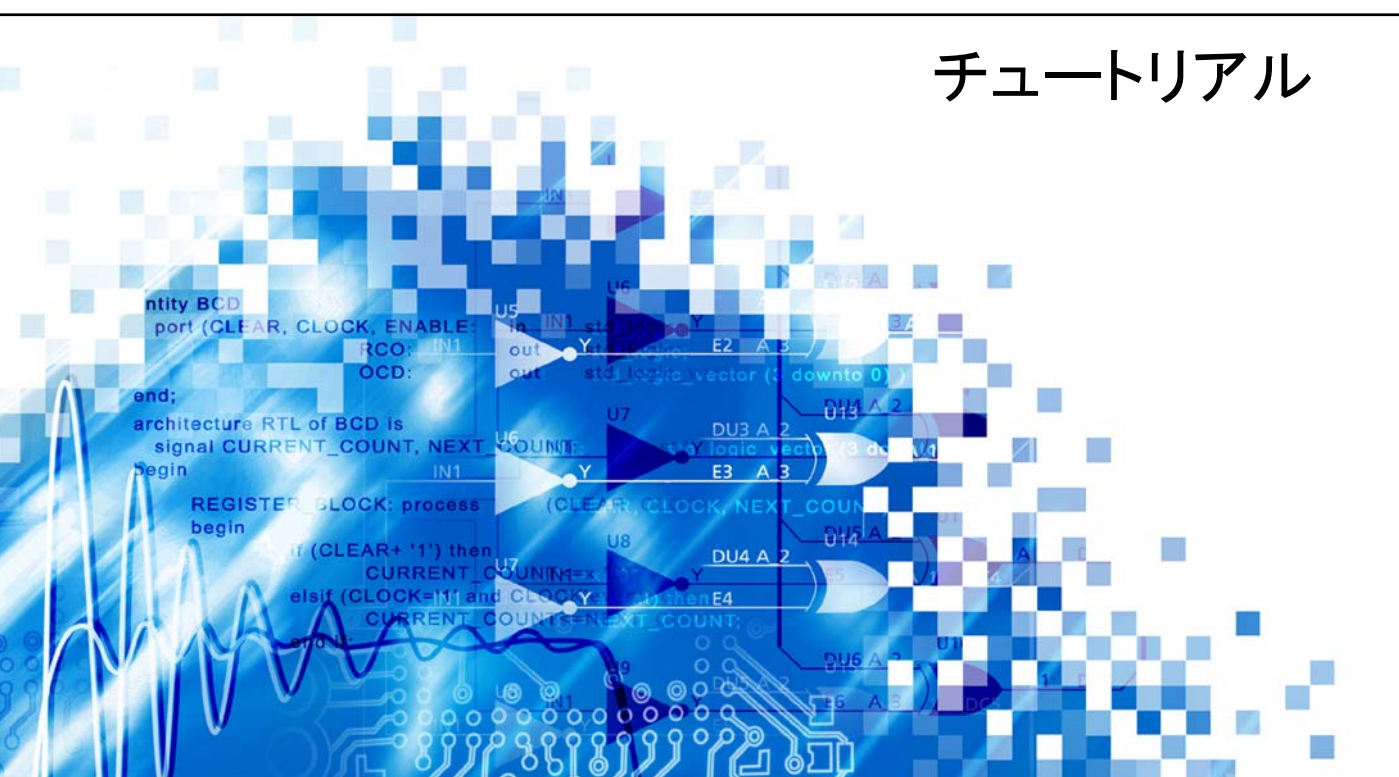


# VHDL と回路図キャプチャー

## チュートリアル



Software, documentation and related materials:

Copyright © 2002 Altium Limited.

All rights reserved. Unauthorized duplication, in whole or part, of this document by any means, mechanical or electronic, including translation into another language, except for brief excerpts in published reviews, is prohibited without the express written permission of Altium Limited.

Unauthorized duplication of this work may also be prohibited by local statute. Violators may be subject to both criminal and civil penalties, including fines and/or imprisonment.

Altium, Protel, Protel DXP, DXP, Design Explorer, nVisage, CAMtastic!, Situs and Topological Autorouting and their respective logos are trademarks or registered trademarks of Altium Limited. All other registered or unregistered trademarks referenced herein are the property of their respective owners and no trademark rights to the same are claimed.

## DXP での VHDL と回路図キャプチャー

DXP のマルチ・ディメンショナルキャプチャー .....	1
FPGA プロジェクトの作成 .....	2
プロジェクトへ VHDL ドキュメントを追加 .....	2
VHDL ドキュメントの新規作成 .....	2
VHDL のトップ階層回路図作成 .....	3
VHDL ファイルからシートシンボルを作成する .....	4
回路図コンポーネントの配置 .....	5
回路図へのポートの配置 .....	6
接続の作成 .....	6
バスの配線 .....	7
ネットラベルの追加 .....	7
VHDL テストベンチファイルの追加 .....	7
VHDL テストベンチの新規作成 .....	8
VHDL モデルの追加 .....	8
VHDL モデルドキュメントの新規作成 .....	8
VHDL ライブラリの使用 .....	9
VHDL ライブラリドキュメントの新規作成 .....	10
プロジェクトのセットアップ .....	11
デザインのコンパイル .....	12
Smart Compile .....	13
デザインのシミュレーション .....	13
デバッグモード .....	15
ブレークポイントの設定 .....	15
シミュレーションの実行 .....	16
ウォッチ・ポイントの設定 .....	16

## DXP のマルチ・ディメンショナルキャプチャー

このチュートリアルでは、DXP を使用して回路図と VHDL の混在したデザインを作成し、シミュレーションを実行するまでの一連の工程を説明します。このチュートリアルは、VHDL コードとテストベンチファイルを作成する知識がある方向けに書かれており、これ以降 FPGA プロジェクトの作成、FPGA ソースの作成、必要となる回路図とテストベンチドキュメントの準備、VHDL モデルと VHDL ライブラリの使用、VHDL シミュレーションを実行し Waveform Viewer で波形を解析するまでを見ていきます。なお、このチュートリアルで使用するサンプルは \Altium\Examples\FPGA\BCD Counter フォルダにあります。

## FPGA プロジェクトの作成

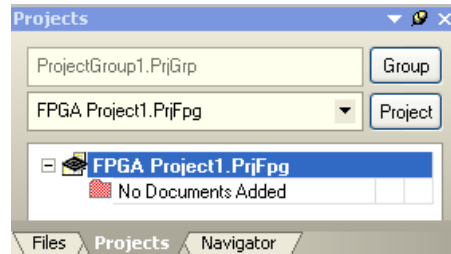
チュートリアル最初に FPGA プロジェクトを作成します：

1. デザインウィンドウの **Pick a Task** セクション内の **Create a new FPGA Design Project** をクリックします。



あるいは、メニューから **File » New » FPGA Project** を選択するかまたは、**Files** パネルの **New** セクション内の **Blank Project (FPGA)** をクリックします。**Files** パネルが表示されていない場合は、パネル下部の **Files** タブをクリックして表示します。

2. 作成した FPGA Project1.PrjFpg が、**Projects** パネルに **No Documents Added** と表示されてリストに加えられています。
3. メニューから **File » Save Project As** を選択して、作成したプロジェクトファイルの名称（拡張子.PrjFpg は付けたまま）を変更します。プロジェクトを保存したいハードディスクの場所を指定し、**ファイル名(N)** フィールドに変更するプロジェクトファイルの名称 BCD.PrjFpg と入力して **保存(S)** ボタンをクリックします



## プロジェクトへ VHDL ドキュメントを追加

このチュートリアルでは、4 ビット BCD カウンターの VHDL コードソースファイル `\Altium\Examples\FPGA\BCD Counter\BCD.VHD` を追加することにします。

1. **Projects** パネルのプロジェクト名の上で、マウスを右クリックして **Add to Project** を選択すれば VHDL ファイルをプロジェクトに追加できます。ファイルの場所とファイル名を指定してから **開く(O)** ボタンをクリックします。
2. 追加された VHDL ドキュメントは、**Projects** パネルの **VHDL Documents** の下に表示されます。ファイル名をダブルクリックするとテキストエディタが開き、このファイルを編集することができます。
3. プロジェクトをセーブします (**File » Save**)。

VHDL に論理合成されないコードを使用するには、以下のコードの間にプログラムを記述します。

```
-- rtl_synthesis off
-- rtl_synthesis on
こちらも使用できます
-- synopsys translate_off
-- synopsys translate_on
```

## VHDL ドキュメントの新規作成

新しい VHDL ドキュメントを作成してプロジェクトに追加したい場合には：

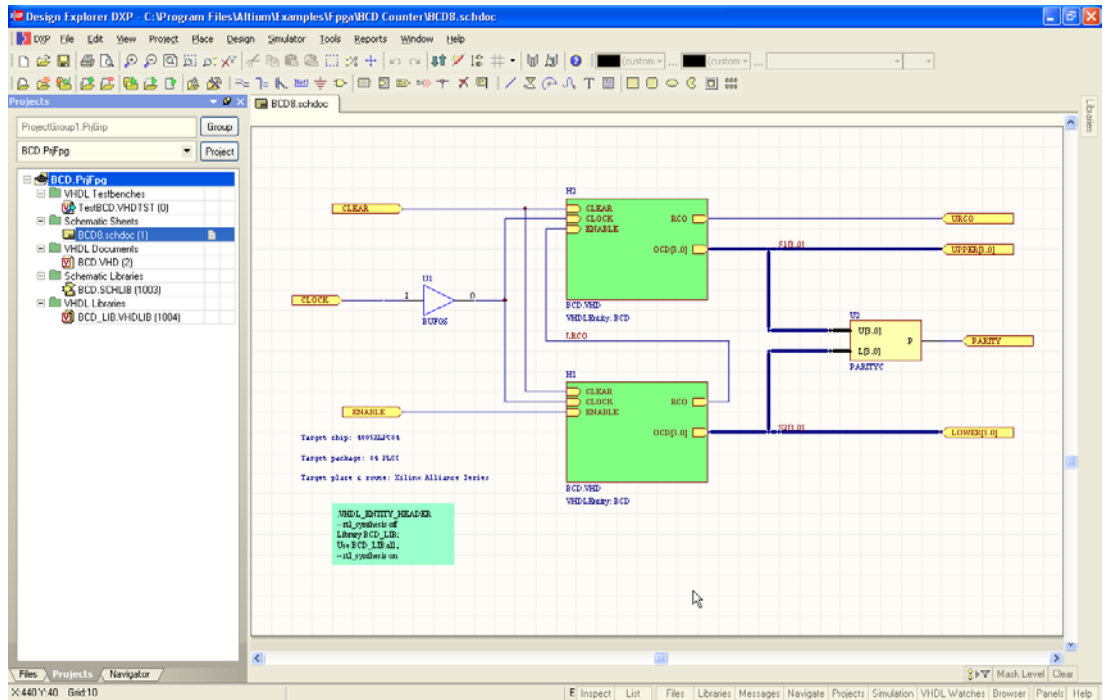
1. メニューから **Select File » New » VHDL Document** を選択すると、新規 VHDL ドキュメント `VHDL1.Vhd` が **Projects** パネルの **VHDL Documents** フォルダ下に作成されます。同時にテキストエディタが開き、すぐに VHDL コードを入力することができます。
2. 必要なコードを入力した VHDL ファイルを.VHD 拡張子付きでセーブ (**File » Save As**) します。
3. プロジェクトをセーブします (**File » Save**)。

次は、BCD カウンターを回路図入力を入力し、プロジェクトファイルに追加します。

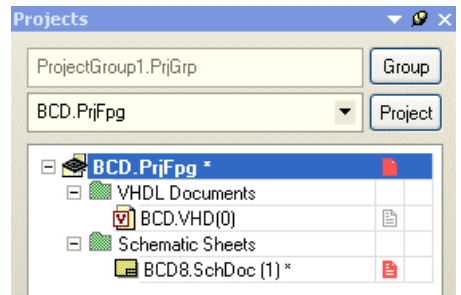
## VHDL のトップ階層回路図作成

FPGA プロジェクトでは回路図および VHDL の 2 タイプのソースドキュメントをサポートしています。両タイプがプロジェクトに混在することも可能です、しかし VHDL でのソースでは、構造化 VHDL で作成される必要があります。シートシンボルを介して、同一プロジェクトに両タイプのドキュメントを混在させることが可能です。

BCD カウンタープロジェクトでは、1 枚の回路図を作成する必要があるため、下の例に示した回路図 BCD8.SchDoc を作成します。



1. メニューから **File » New » Schematic** を選択するか、または **Files** パネルの **New** セクションにある **Schematic Sheet** をクリックすると、空白の回路図 Sheet1.SchDoc がデザインウィンドウで表示されます。
2. **File » Save As** を選択し、新たに作成された回路図のファイル名を変更して (拡張子 .SchDoc は付けたまま) セーブします。開いた **Save** ダイアログで、回路図を保存するフォルダを指定して、**ファイル名 (N)** のフィールドに BCD8.SchDoc と入力して **保存 (S)** をクリックします。
3. 回路図のテンプレートを読み込むには、**Design » Template » Set Template File Name** を選択し、開いたダイアログで \Altium\Templates

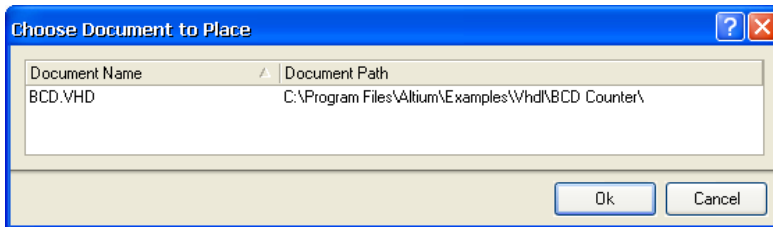


フォルダにある A4.SchDot を選んで **開く(O)** ボタンをクリックします。テンプレートがセットされるまで各ダイアログに **OK** をクリックし、回路図をセーブします。

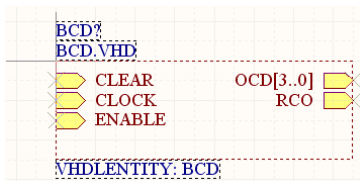
### VHDL ファイルからシートシンボルを作成する

シートシンボルを使うと、VHDL と回路図の混在使用が可能です。VHDL の場合、シートシンボルに生成されたシートエントリが VHDL ドキュメントの各ポートに対応しています。

1. デザインウィンドウに新規回路図が表示されている状態で、メニューから **Design » Create Symbol From Sheet** を選択すると、ドキュメントからシートシンボルが生成されます。今回の例では、VHDL ドキュメント BCD.VHD 中のエンティティ BCD のために、シートシンボル H1 を生成します。
2. 表示している **Choose Document to Place** ダイアログで、BCD.VHD を指定して **OK** ボタンをクリックします。

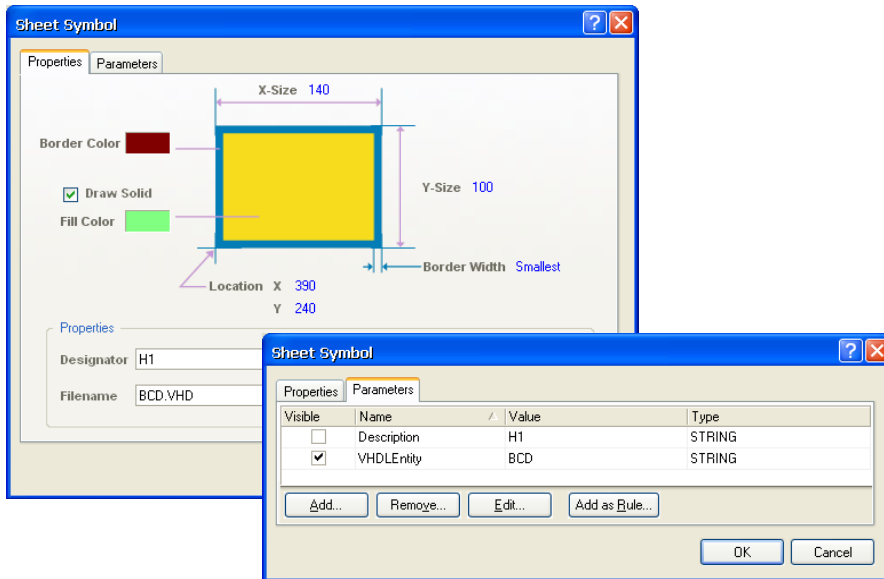


3. 今回のケースでは VHDL エンティティは1つでしたが、VHDL ドキュメントに複数のエンティティが存在する場合は、**Choose VHDL Entity** ダイアログが表示されるので、そこから1つのエンティティを指定します。以上の操作でカーソル上にシートシンボルが表示されます。

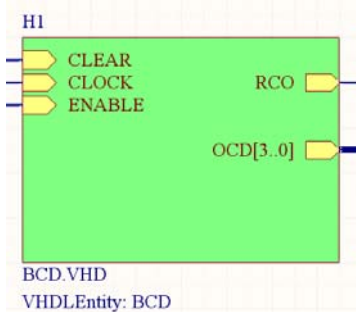


VHDL ファイルにあるエンティティに正しくリンクするために、VHDL Entity パラメータは、シートシンボルに常に存在する必要があります。これは VHDL ファイルが複数のエンティティを含むことができるからです。

4. シートシンボルを回路図に配置する前に **TAB** キーを押すとシートシンボルのプロパティを、**Designator** を H1 にセットするなどの設定できます。**Parameters** タブをクリックすると、自動的に生成された VHDL ENTITY パラメータがすでにセットされており、他の **description** といった必要なパラメータも追加することができます。



5. 回路図にシートシンボルを配置したら、下図を参考にしてシートエントリやテキストの位置を変更します。



ターゲットファイルがソースドキュメントと思われるときは、シートシンボルを使用すべきです。ターゲット VHDL ファイルが、デザインのライブラリドキュメントと思われるときは、モデルとライブラリを使用すべきです。

6. 1 から 5 までを繰り返し 2 個目のシートシンボルを、Designator を H2 と設定してから配置し、回路図をセーブします。

## 回路図コンポーネントの配置

デザインに回路図シンボルを使用することもできますが、その場合には、そのコンポーネントの VHDL ピヘイビアモデルが、モデルファイルか VHDL ライブラリに記述されていることが必要です。今回はその両方を使用した例を示すために、まず初めにコンポーネントを配置します。このデザインで使用するコンポーネントを含んだ回路図ライブラリは `\Altium\Examples\FPGA\BCD Counter\SCH Library\BCD.SchLib` にあります。

1. 回路図ライブラリを追加するには、**Projects** パネルの BCD.PrjFpg 名前

統合ライブラリは、すべてのモデルを含んだ 1 つの大きなファイルか、それぞれ 1 つの実体を含む個別ファイルとしての VHDL モデルをサポートしています。

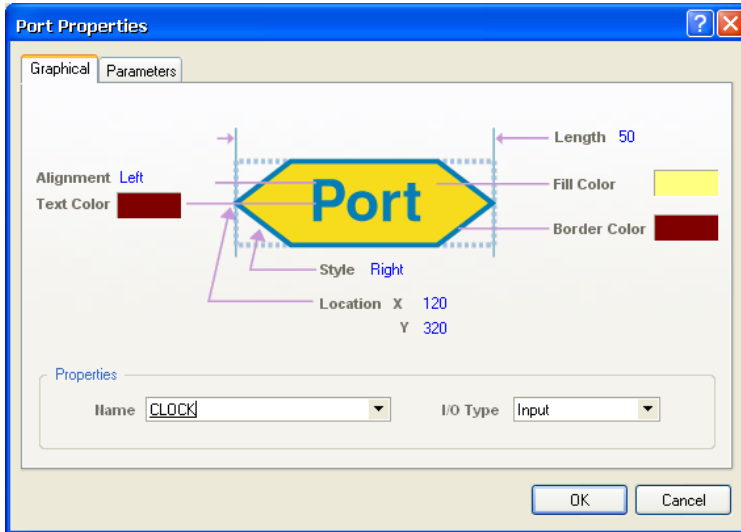
部分を右クリックして **Add To Project** を選択し、BCD.SCHLIB を指定して **開く(O)** をクリックします。

- 次に 2 つのコンポーネントを配置します。メニューから **Place » Part** [ショートカット P, P] を選択し、開いた **Place Part** ダイアログの **Lib Ref** フィールドに BUFGS と入力してバッファコンポーネントを、同様に **Lib Ref** フィールドに PARITYC と入力してパリティコンポーネントを、上の回路図を参考にして配置します。

### 回路図へのポートの配置

Clock インポートポートを配置します：

- メニューから **Place » Port** [ショートカット P, R] を選択します。カーソルの形状が変りポートのアウトラインがカーソル上に表示されています、**TAB** キーを押して **Port Properties** ダイアログを表示させます。**Name** フィールドに **Clock** と入力し、**I/O Type** に **Input** を指定します。ポートシンボル形状を変更したい場合は **Style** を **Right** に変更して **OK** ボタンを押します。これらの設定は、他のポートを配置する際にも有効になります。



- ポートの配置位置をクリックし、マウスを移動してポートの長さを決定してから再度クリックして配置を完了します。
- 1 と 2 を繰り返して、他の入力ポートを配置します。
- バスの信号線 Lower[3..0] のポートを、**Port Properties** ダイアログで **I/O Type** を **Output** に **Style** を **Left** に設定し **Name** フィールドに LOWER[3..0] と入力してから **OK** ボタンを押してポートを配置します。
- マウスを右クリックするか、または **ESC** キーを押してポート配置モードを終了します。 .

### 接続の作成

以上で、回路図の接続が開始できます。



1. ワイヤを配置するには、メニューから **Place » Wire** [ショートカット P, W] を選択し、ワイヤを引き始めるポイント（通常はポートやコンポーネントのピン）でクリックします。カーソルを移動させ、接続したい次のポイントで再度クリックします。他のポートやコンポーネントのピンまでの、接続が完了するまで繰り返し、マウス右クリックで終了します。ワイヤ配置モードを終了するにはマウス右クリックか **ESC** キーを押します。
2. 回路図ファイル BCD8.SchDoc を参考にして配線を引いていきます、その際ワイヤが交差している位置でのジャンクションの有無に注意してください。もし 2 本のワイヤが交差しそこにジャンクションがあるならば、この 2 本のワイヤは接続されます。もしジャンクションが無ければ、これらは接続されていません。この回路図では、ワイヤが接続されるポイントには自動的にジャンクションが配置されます。

## バスの配線

バスから他のオブジェクトへ接続する場合は、バスメンバーは左から右へと決定されます、またバスのメンバーの数は両者同一でなければなりません。

ポート LOWER[3..0]へバスを接続します：

1. メニューから **Place » Bus** [ショートカット P, B] を選択し、バス配置モードにします。
2. ポート LOWER[3..0]とシートエントリ OCD[3..0]間を、ワイヤを配置したときと同じやり方でバスを配置します。

DXP では確実な接続を行なうため、バスにはネットラベルが必要です。

## ネットラベルの追加

接続にネットラベルを付けておくのは、後にデバックを簡単に行なえる様になる良いやり方です。もしネットラベルが無ければ、DXP はネットラベルを出力時に生成します。今回の例ではネットラベルをネット LRCO にだけ付け加えます。

1. メニューから **Place » Net Label** [ショートカット P, N] を選択すると点線に囲まれたラベルが、カーソルの上に表示されます。
2. ネットラベルを配置する前に **TAB** キーを押すと、表示した *Net Label* ダイアログでネットラベルのプロパティが編集できます。ネットラベル名 LRCO を **Net** フィールドに入力して **OK** ボタンをクリックします。
3. カーソルを、ネットラベルの文字の左下（ホットスポットと呼びます）が、目的のワイヤに触る位置まで移動させると、カーソルが赤いクロスに変わるので、その位置にネットラベルを配置します。
4. 配置が終了したら、マウスを右クリックするか **ESC** キーを押してネットラベル配置モードを終了します。
5. **File » Save** を選択して回路図をセーブします。

**Design » Create Sheet From Symbol**, や **Design » Create VHDL File From Symbol** コマンドを使うと、VHDL ファイルの中のコンポーネントをすぐさま別のシートで宣言 (declare) や例示 (instantiate) することができます。

## VHDL テストベンチファイルの追加

テストベンチファイルとは、デザインの実行をテストするシーケンスを記述した VHDL ソースファイルです。テストベンチファイルはコア・デザインの一部ではないので、デザインの階層やネットリス

トには含まれません。テストベンチはシミュレートするデザインの最上層に存在されるべきファイルです。

このチュートリアルでは、\Altium\Examples\FPGA\BCD Counter\BCD.VHDTST にあるテストベンチを、このプロジェクトで追加して使用します。

1. **Projects** パネルのプロジェクト名称の上で右クリックし、**Add to Project** を選択しファイルを指定してから **開く(O)** ボタンを押して、この VHDL ファイルを FPGA プロジェクトに追加します。
2. VHDL テストベンチドキュメントがプロジェクトに追加され、**Projects** パネルの VHDL Testbenches フォルダ下にリストされます。ファイル名をダブルクリックするとテキストエディタで表示され編集ができます。
3. プロジェクトをセーブします (**File » Save**)。

### VHDL テストベンチの新規作成

新しい VHDL テストベンチドキュメントを作成してプロジェクトに追加したい場合は：

1. **Files** パネルの **New** セクションの **Other Document » VHDL Testbench** を選択します。
2. <name>.VHDTST、この場合 BCD.VHDTST ファイルが作成され、**Projects** パネルの VHDL Testbenches フォルダ下にリストされます。
3. 作成したテストベンチファイルをオープンし、必要なテストベンチコードを入力してから、このファイルをセーブします。

### VHDL モデルの追加

---

デザインの作成はほとんど終了しましたが、まだシミュレーション用のビヘイビアモデルを含んでいない2つのコンポーネントがあります。この BUFGS コンポーネントにモデルを追加します。

このチュートリアルでは BUFGS コンポーネントの VHDL ビヘイビアモデルを含んだ、\Altium\Examples\FPGA\BCD Counter\VHDL Models\BUFGS.VHMDL を追加してみます。

1. **Projects** パネルのプロジェクト名称の上で右クリックし、**Add to Project** を選択しファイルを指定してから **開く(O)** ボタンを押して、この VHDL ファイルを FPGA プロジェクトに追加します。
2. VHDL モデルドキュメントがプロジェクトに追加され、**Projects** パネルの VHDL Model Libraries フォルダ下にリストされます。ファイル名をダブルクリックするとテキストエディタで表示され編集ができます。
3. プロジェクトをセーブします (**File » Save**)。

### VHDL モデルドキュメントの新規作成

新しい VHDL モデルドキュメントを作成してプロジェクトに追加したい場合は：

1. **Files** パネルの **New** セクションの **Other Document » VHDL Testbench** を選択します。
2. <name>.VHMDL のフォーマット、この場合では BUFGS.VHMDL のファイル名でセーブします。セーブされたファイルは **Projects** パネルの VHDL Model Libraries フォルダ下にリストされます。

3. 作成した VHDL モデルファイルをオープンし、このコンポーネントのエンティティとアーキテクチャーコードを入力してから、ファイルをセーブします。

この VHDL モデルを BUFGS コンポーネントに追加します。シミュレーターはそのコンポーネントのモデルを検出して、コンパイルプロセスに組み込みます。これを行なうには：

1. 回路図シート上の BUFGS コンポーネントをダブルクリックし **Component Properties** ダイアログを表示させます。
2. ダイアログ右下にある **Models** セクションの **Add** ボタンを押して、**Add New Model** ダイアログを表示します。そのドロップダウンリストから VHDL を指定して **OK** ボタンを押すと、**VHDL Model Properties** ダイアログが開きます。
3. モデルの名前は、VHDL モデルファイルの中のエンティティと同じでなければならないので、BUFGS と入力します。このコンポーネントは統合ライブラリのオリジナルではないので、**From integrated library** ラジオボタンはグレーアウトしています。すでにモデルファイルは FPGA プロジェクトの追加されていますので、場所を指定する必要はありません。あるいは直接 **Options for Project dialog (Project » Project Options)** ダイアログの **Search Paths** タブに、フルパスで `Altium\Examples\FPGA\BCD Counter\VHDL Models\` と入力しておくこともできます。
4. **Description** フィールドには BUFGS と入力します。この特殊なモデルは、シミュレーションとシンセシスの両方に使用できるモデルですので、**Simulation/Synthesis** ボックスには **Simulation and Synthesis** と表示されています。**OK** を押してダイアログを閉じます。

## VHDL ライブラリの使用

PARITY コンポーネントは VHDL ライブラリを利用します。VHDL ライブラリには、様々なコンポーネント、パッケージ、タイプ、定数などを含んだ VHDL ドキュメントが含まれます。これらのファイルは、VHDL ライブラリファイルの名称と同じ論理名にマップされています。

VHDL コード内で **Library** と **Use** ステートメントを使用することで、このライブラリからエクスポートされたすべての VHDL ファイルの構造を利用することができます。シミュレーターは、最初にすべての VHDL ライブラリをコンパイルし、続いてプロジェクトのモデルファイルとソースドキュメントをコンパイルします。

このチュートリアルでは **Parity** コンポーネントのビヘイビアコードをエクスポートした VHDL ライブラリファイル `Altium\Examples\FPGA\BCD Counter\VHDL Library\BCD_LIB.VHDLIB` を追加します。

1. FPGA プロジェクトに VHDL ライブラリを追加するには、**Projects** パネルのプロジェクト名の上で右クリックし **Add to Project** を選択、ファイルを指定して **開く(O)** ボタンを押します。
2. VHDL ライブラリドキュメントはプロジェクトに追加され、**Projects** パネルの **VHDL Libraries** フォルダの下にリストされます。ファイル名をダブルクリックすると、テキストエディタが開きファイルの閲覧や編集ができます。
3. プロジェクトをセーブします (**File » Save**)

回路図に配置できる VHDL コードのキーワードは以下の通りです。

```
.VHDL_ENTITY_HEADER
.VHDL_ENTITY_GENERIC
.VHDL_ENTITY_DECLARATION
.VHDL_ENTITY_STATEMENT
.VHDL_ARCH_HEADER
.VHDL_ARCH_DECLARATION
.VHDL_STATEMENT
```

## VHDL ライブラリドキュメントの新規作成

新しい VHDL ライブラリドキュメントを作成して、プロジェクトに追加する場合は：

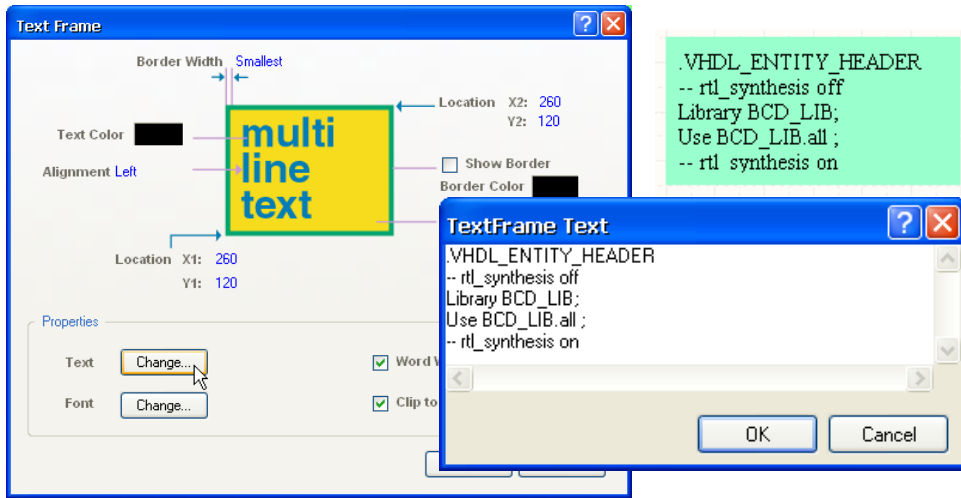
1. Files パネルの New セクションから **Other Document » VHDL Library** を選択します。
2. <name>.VHDLIB のフォーマット、この場合では BCD\_LIB.VHDLIB のファイル名でセーブします。セーブされたファイルは **Projects** パネルの VHDL Libraries フォルダ下にリストされます。
3. **VHDL » Edit Library** を選択すると、ライブラリのドキュメントを追加できる *Edit VHDL Library* ダイアログが開きます。

VHDL ライブラリを作成したかまたは既存のライブラリを使用したかにかかわらず、以下のステップを行なってください：

1. BCD\_LIB.VHDLIB ドキュメントがオープンされていることを確認してから、**VHDL » Edit Library** を選択すると、*Edit VHDL Library* ダイアログが開き PARITY.VHD と UTILITY.VHD の 2 つのファイルが表示されています。最初のファイルは、**Parity** コンポーネントのためのエンティティ宣言を含んでいます、2 番目のファイルには、パリティ機能を持つ **Utility** と呼ばれるパッケージが入っています。ライブラリをコンパイルする順序は下から順に行なわれます、この場合、PARITY.VHD の前に UTILITY.VHD がコンパイルされます。
2. ファイルの並びが適切な順序であることを確認して **OK** ボタンを押します。
3. **VHDL » Edit Library** を選択すると、ライブラリのドキュメントを追加できる *Edit VHDL Library* ダイアログが開きます。

プロジェクトに BCD.VHDLIB ライブラリドキュメントを追加しましたが、デザインで適切なリファレンスを付ける必要があります。VHDL ファイルでこれらを行なうには **Library** と **Uses** ステートメントを使います。しかし回路図でコンポーネントを使用しているときは、以下のことを行なう必要があります：

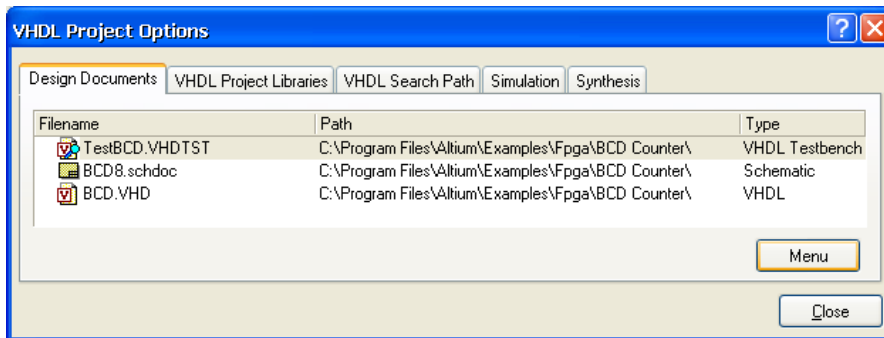
1. BCD8.SchDoc ドキュメントがオープンされていることを確認してから、**Place » Text Frame** を選択します。回路図の左下に移動しそこにテキストフレームを配置してから、ダブルクリックして *Text Frame* ダイアログを表示させます。Text のとなりの **Change** ボタンを押して *TextFrame Text* ダイアログを表示します。
2. 回路図の中で、VHDL コードの特別なヘッダを挿入することができます。この場合 **Library** ステートメントを挿入したいので、ライブラリステートメントの宣言.VHDL\_ENTITY\_HEADER を記述します。下に示したテキストを入力し、**OK** ボタンをクリックします。



## プロジェクトのセットアップ

トップ階層の回路図と VHDL ファイルをプロジェクトに追加したので、プロジェクトのオプションを設定します。

1. FPGA プロジェクトに追加してある VHDL または回路図、例えば BCD.VHD をオープンします。
2. メニューから **Design » FPGA Options** を選択し、開いた *VHDL Project Options* ダイアログで設定します。

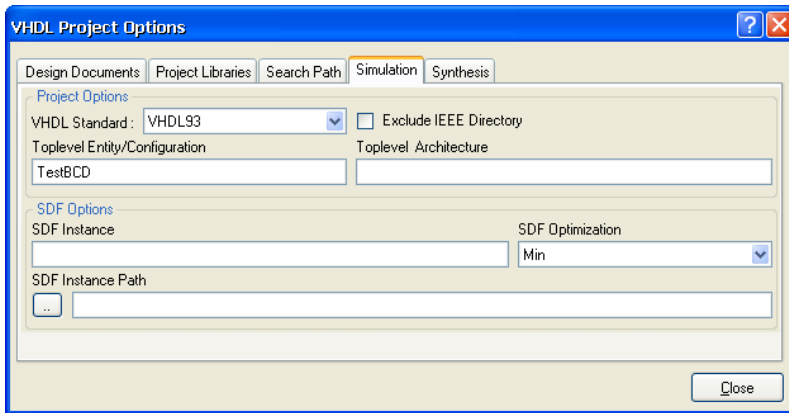


3. **Design Documents** タブには、このプロジェクトに含まれるソースドキュメントのリストが表示されています。ドキュメントは下から上の順にコンパイルされます、この場合では BCD.VHD が最初にコンパイルされ次に回路図、最後にテストベンチファイルの順となります。ドラッグアンドドロップまたは **Compile Sooner** と **Compile Later** (**Menu** ボタンをクリックして表示) を使えばドキュメントの順序を、上の図のように変更できます。DXP » **System Preferences** で開く *Design Explorer Preference* ダイアログ Projects Panel タブ内の Show document position in project にチェックマ

コンパイルされる順序で Projects パネルのファイルが表示されるようにするには、DXP » **System Preferences** を選択し **Projects Panel** タブ内の Sort By セクションで設定できます。

ークを付けて数字を表示するように設定した場合には、**Projects** パネルのドキュメント隣の数字にもドキュメントの順序が反映されます。

- VHDL Project Libraries** タブには、このプロジェクトに含まれているライブラリのリストが表示されています。この例ではライブラリは1つしか含まれていませんが、ライブラリは下から順にビルドされます。そのためリストの最後のライブラリが最初にコンパイルされ、リスト一番上のライブラリが最後にコンパイルされることになります。**Menu** ボタンをクリックするか、ライブラリ名称の部分で右クリックすると、ライブラリに適応できるコマンドを見ることができます。
- Simulation** タブをクリックして **Top-level entity/configuration** フィールドにエンティティの名称を入力します、この例ではテストベンチファイル BCD.VHDTST で定義されている通り、トップ階層のエンティティは TestBCD です。



## デザインのコンパイル

以上でデザインをコンパイルする準備が整いました。ライブラリファイル、モデルファイル、ソースファイルの順番でコンパイルされます。回路図ファイルは VHDL に変換され、対応する VHDL ファイルはシミュレーターによってコンパイルされます。

- プロジェクトに含まれたソースドキュメントどれかをオープンした状態で、メニューから **Simulator » VHDL Compile** を選択すると、**Project Outputs** フォルダに回路図ドキュメントから変換された BCD8.VHD が作成され、**Projects** パネルの **Generated VHDL Documents** フォルダ下に表示されます。
- メニューから **View » Workspace Panels » Messages** を選択するか、**Messages** タブをクリックして **Messages** パネルを開き、コンパイル中に発生したかもしれないエラーを確認してください。もし **Messages** パネルにエラーかウォーニングが表示されていたなら、エラー行をダブルクリックすることで、その原因となる VHDL コードの該当行または回路図上の該当部分へジャンプすることができます。
- VHDL ファイルまたは回路図を修正し、プロジェクトファイルをセーブします。

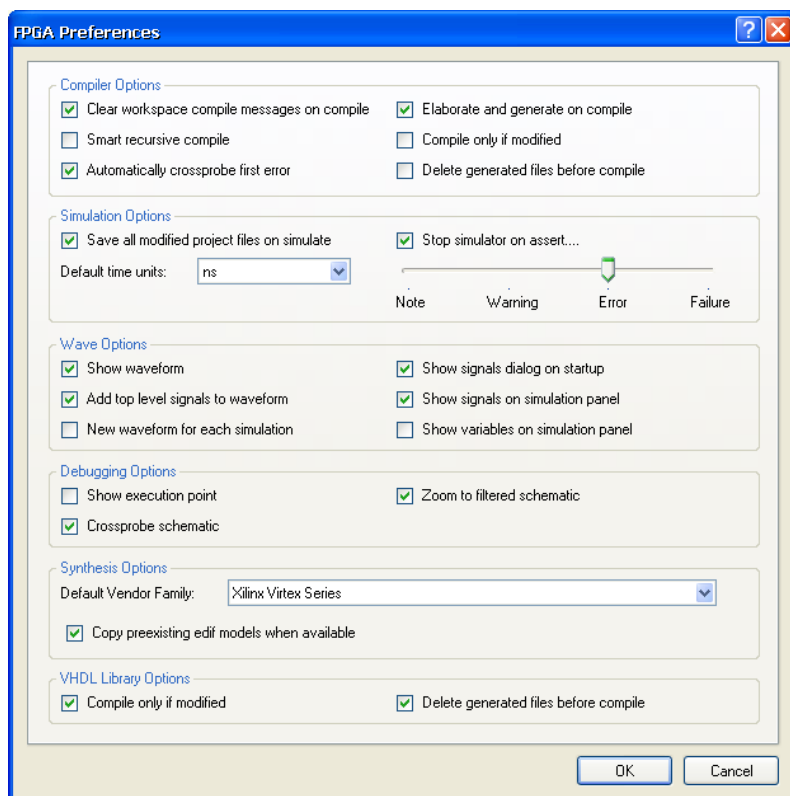
## Smart Compile

コンパイラーには、コンパイルする際の順序を自動的に決定するオプションがあります。このオプションは'Smart Compile'と呼ばれ、*FPGA Preferences* ダイアログ(**Tools » FPGA Preferences**)の **Smart recursive compile** オプションをオンにすることで機能します。ボトムアップの順序でデザインが全てコンパイルされるか、これ以上コンパイルされるファイルが無くなるまで再帰的にコンパイルを続行します。プロセスの最後に、コンパイルの順序を調整することができます。

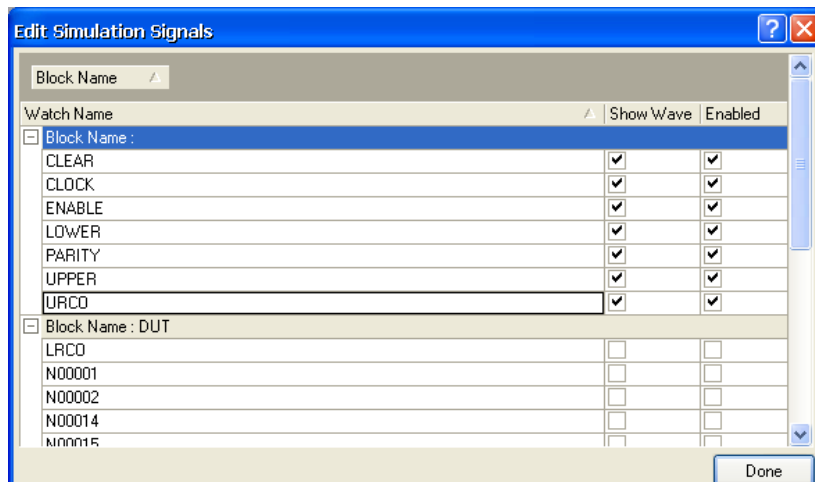
## デザインのシミュレーション

以上で、デザインのシミュレーションを始めることができます。シミュレーションを開始する前に、ソースまたはライブラリドキュメントに変更があった場合、プリファレンスの設定を変更する必要があるかもしれません。

1. テストベンチファイル<name>.VHDTST、この場合は BCD.VHDTST をオープンします。
2. **Design » FPGA Options** を選択し、*VHDL Project Options* ダイアログの設定をチェックまたは変更します。
3. **Tools » FPGA Preferences** を選択し、*FPGA Preferences* ダイアログの設定をチェックまたは変更します。**Show Waveform** や **Add top level signals to Waveform** といった **Wave Options** が選択されていれば、**Waveform Viewer** が自動的にオープンします。確認して **OK** ボタンをクリックします。

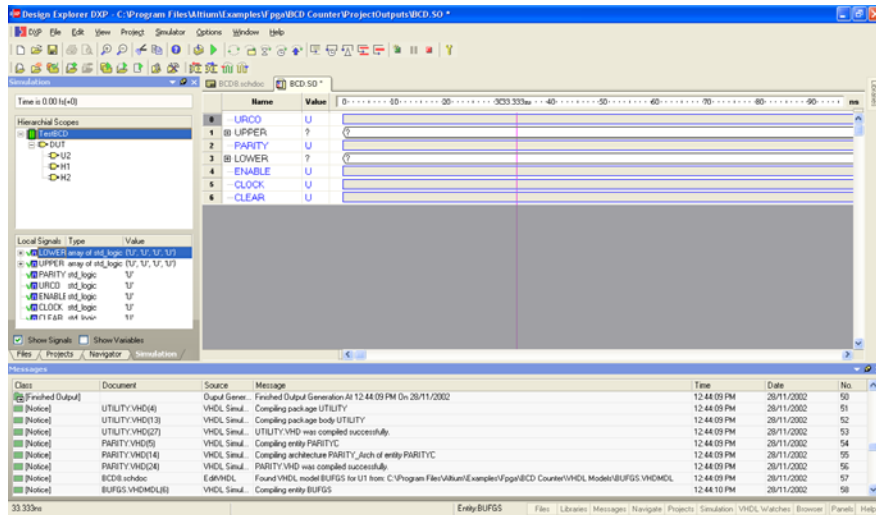


4. **Simulation** タブをクリックして **Simulation** パネルを表示します。
5. **Simulator** » **Simulate** を選択すると、シミュレーションがイニシャライズされ、*Edit Simulation Signals* ダイアログが表示されます。





6. どのシミュレーション波形を、表示あるいはイネーブルにするかを選択してください。イネーブルにすると、信号の推移をトラッキングします、**Show Wave** だけ選択しているときには、波形は観察できますが時間の推移はセーブされません。**Done** ボタンをクリックすると空白の .SO ファイルが作成され **Waveform Viewer** が開きます。ドキュメントタブをクリックすると表示されます。



‘U’は‘Undefined’を意味することに注意してください。これらの値はシミュレーションを実行したときに変更されます。

7. シミュレーションがイニシャライズされたならば、テストベンチの中で指定された時間の終わりまでシミュレーションを実行するか、またはデバッグモードを開始できます。

## デバッグモード

デバッグモードの時、VHDL メニューから **Custom Step (Simulator » Custom Step)**, **Step Time**, **Delta Step**, **Step Into** または **Step Over** といった、ステップバイステップのシミュレーションオプションが使用できます。現在の実行ポイントを確認したい場合は **FPGA Preferences** ダイアログの **Show Execution Point** オプションをチェックしてください。

## ブレイクポイントの設定

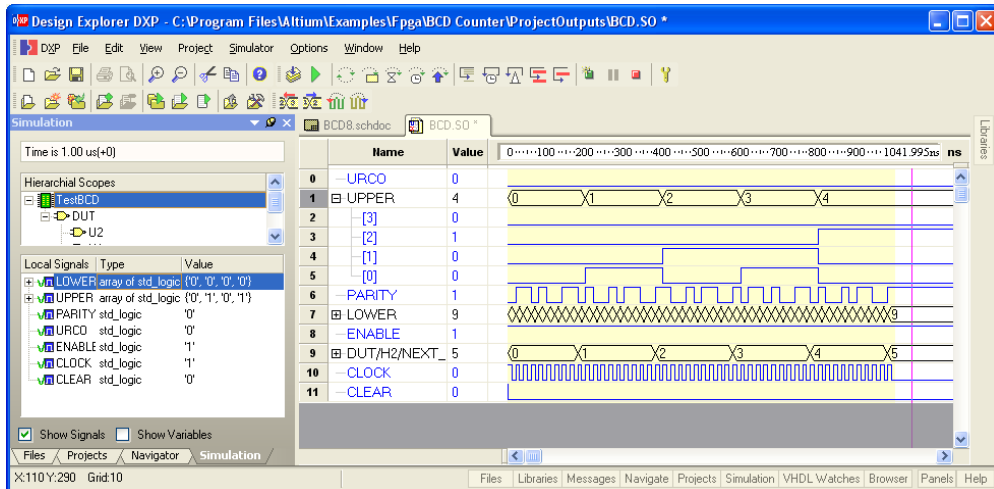
ブレイクポイントの設定は：VHDL ファイルが開かれている状態で、ブレイクポイントが設定できるラインに、青い小さな点が左側マージン部分に表示されています。

- ブレイクポイントを設定したい行の横のマージン部分をクリックすると、X マークが表示された赤い点で示された、ブレイクポイントマークが表示されます。
- ブレイクポイントを解除するには、再度ブレイクポイントマークをクリックします。
- Breakpoints** タブをクリックするとブレイクポイント・データが表示された **Breakpoints** パネルが開きます。

## シミュレーションの実行

シミュレーションがセットアップされると、すべての Run オプションが **Simulator** メニューから利用可能になります。

1. 適切な実行命令、例えば **Simulator » Run** を選択すると *Enter Time Step* ダイアログが表示されるので、シミュレーションの実行時間を入力して **OK** ボタンをクリックします。入力した実行時間は **Simulator » Run for <time step>** コマンドのためのデフォルト時間になります。
2. シミュレーションが実行され、実行中には VHDL メニューの、**Stop**、**End** と **Restart** コマンドが使用できます。
3. シミュレーションの結果は、**Simulation** パネル、**Waveform Viewer**、**Messages** パネルで確認することができます。**Simulation** パネルでは VHDL デザインの構造と有効な信号の変数と型、シミュレーション・タイム時の値を確認することができます。



Waveform Viewer では各イベント間の時間の計測 (**Edit » Measure Time**)、遷移点または time marks を設定したポイントへのジャンプ (**Edit » Jump**)、信号のグループ化 (**Edit » Insert » Signal Group**)、信号の比較 (**Edit » Insert » Comparison**) が行なえます。

## ウォッチ・ポイントの設定

VHDL Watches パネルを表示すると、ウォッチ・ポイントを設定することができます。

1. **Simulation** パネルから **VHDL Watches** パネルへ、信号をドラッグアンドドロップします。あるいは右クリックメニューからでも同様の操作ができます。
2. **VHDL Watches** タブをクリックして **VHDL Watches** パネルを表示すれば、設定したウォッチ・ポイントの確認ができます。

The screenshot shows the Design Explorer DXP interface during a simulation. The main window displays a signal list with the following data:

Name	Value
0 DUT/H2/NEXT_?	?
1 URCD	U
2 UPPER	?
3 PARITY	U
4 LOWER	?
5 ENABLE	U
6 CLOCK	U
7 CLEAR	U

The VHDL Watches window shows the following data:

Name	Type	Value	Last Value
UPPER	array of std_logic	{'U', 'U', 'U', 'U'}	{'U', 'U', 'U', 'U'}
PARITY	std_logic	'U'	'U'
URCD	std_logic	'U'	'U'
ENABLE	std_logic	'U'	'U'
CLOCK	std_logic	'U'	'U'
CLEAR	std_logic	'U'	'U'