



FPGA デザイン入門

概要

Tutorial
TU0116 (v1.1) September 23, 2004

このチュートリアルでは、FPGA 設計を行うための概要を説明します。アルティウムの NanoBoard でドーターボード Xilinx Spartan IIE を使用し、どのようにして回路図を作成し、プログラム、ビルド、論理合成、コンパイルするかという概要を紹介します。また、FPGA 設計において、サブシートと VHDL ファイルの使用法を簡単に紹介します。

このチュートリアルでは、非常に簡単な回路設計を説明します。ソフトウェアを利用する高レベルの回路図/VHDL 混在設計ではありません。もっと高度な設計を行うには、\Altium2004\Examples フォルダのサンプルファイルを参照して下さい。FPGA Hardware フォルダには、プロセッサ（回路図/VHDL 混在マルチチャンネル LED chaser を含む）を含まないファイルが保存されています。FPGA Processors フォルダと Reference Designs フォルダには、プロセッサに基づく FPGA デザイン（デジタルシステム全体を扱う簡単な特徴などを表したファイル）が保存されています。

チュートリアルを始める前に、PC のパラレルポートに NanoBoard を接続し、スイッチを ON にして下さい。Xilinx ウェブサイト(www.Xilinx.com)からダウンロードした Xilinx ツール (web 版) がインストールされていることを確認して下さい。

このチュートリアルでは、図 1 の Johnson Counter のサンプルファイルを使って説明します。このファイルは、DXP のインストールディレクトリの Altium2004\Examples\Tutorials\Getting started with FPGA Design フォルダに保存されています。

Johnson Counter (あるいは Twisted ring counter) は、最終段のフリップフロップの反転した出力が初段のフリップフロップの入力に接続される同期カウンタです。この Johnson counter の回路が動作した時は、NanoBoard 上の LED が一度に点灯します。後から回路図に LED 表示を遅くする為のクロック遅延を追加することで、カウンタは左から右へ、あるいは、右から左へシフトするように見えます。

DXP では、デジタル設計の回路図概念が PCB 設計概念と異なることに注目することは重要です。FPGA 設計では、PCB 設計の規則通りにはいきません。

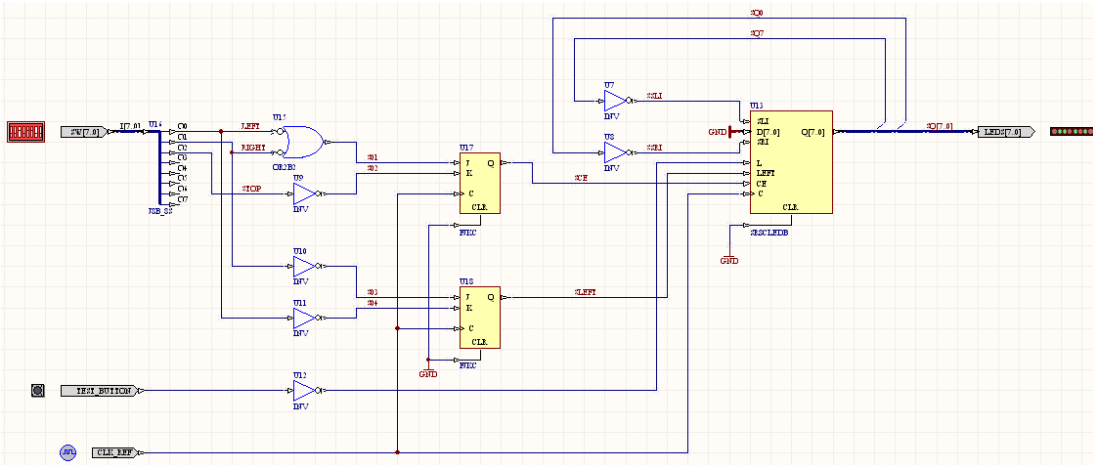


図 1. Johnson Counter 回路図

FPGA プロジェクトの作成

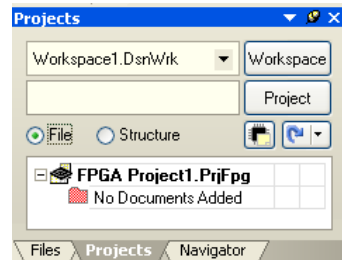
DXP で設計を始める為に、最初にプロジェクトを作成する必要があります。プロジェクトは、個々のドキュメントや出力ファイルを管理する為に使用します。FPGA 設計を行う為に、FPGA プロジェクトを作成する必要があります。

新規の FPGA プロジェクトを作成するには、以下の手順を実行します。

1. メニューから **File » New » Project » FPGA Project** を選択するか、Files パネルの **New** の項目で **Blank Project (FPGA)** をクリックします。もし、このパネルが表示されない場合は、Files タブをクリックして下さい。
2. Projects パネルに新しいプロジェクトファイル (FPGA Project1.PrjFpg) が表示されます。
3. プロジェクトファイル (拡張子 .PrjFpg) の名前を変更するには、**File » Save Project As** を選択します。ファイル名の項目で Johnson_Counter.PrjFpg と入力し、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ます。そして、**保存** をクリックします。

その後の設計過程で論理合成エラーを避ける為に、プロジェクト名、ドキュメント名には、スペースではなくアンダースコア('_')を使用する必要があります。

次に、Johnson Counter の回路図を作成します。



警告: プロジェクト名やファイル名にはスペースやダッシュ(-)は使用しないで下さい。もし、文字間を空けたい場合は、アンダースコアを使用して下さい。

回路図の作成

FPGA プロジェクトは、回路図と VHDL の 2 種類のソースドキュメントをサポートしています。プロジェクトでは、シートシンボルを使用することで両方のドキュメントタイプを混在させることができ

ます。しかし、FPGA プロジェクトでは、プロジェクトのトップ図面で回路図を使用する必要があります。これにより、Vendor FPGA-PCB 統合環境がサポートされます。

デザインには、NanoBoard 上の LED の表示を遅らせる為の遅延回路が含まれています。最初に、Johnson counter の回路図シートを作成します。それから、階層構造の使用例を紹介する為に、サブシートに遅延回路を移動します。最後に、VHDL ファイルでこのサブシートを交換します。

シートシンボルを使用して、VHDL と回路図ドキュメントを混合することができます。VHDL の場合、シートエントリは、VHDL ドキュメントのポートに相当します。

Johnson Counter の回路図を作成するには、以下の手順を実行します。

1. **File » New » Schematic** を選択するか、**Files** パネルの **New** の項目の **Schematic Sheet** をクリックします。デザインウィンドウにブランクの回路図 (Sheet1.SchDoc) が表示されます。
2. 回路図ファイル (拡張子.SchDoc) の名前を変更するには、**File » Save As** を選択します。ファイル名の項目で Johnson_Counter.SchDoc と入力し、回路図を保存したいハードディスクの場所を指定します。そして、**保存** をクリックします。

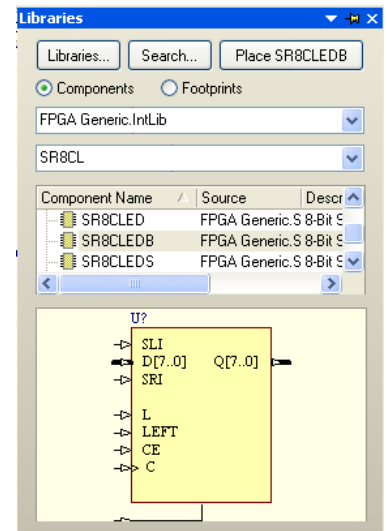
回路図へのパーツ配置

この回路図で使用するコンポーネントは、\Program Files\Altium2004\Library\FPGA\FPGA Generic.IntLib の統合ライブラリに用意されています。このライブラリは、**Libraries** パネルで登録利用することができます。

Generic FPGA ライブラリのコンポーネントは、このシステムでサポートしている全 FPGA デバイスで使用できます。ベンダープリミティブが利用可能な特別な統合ライブラリ(*FPGA.INTLIB)が (\Program Files\Altium2004\Library のデバイスメーカー別フォルダに) あり、これらは特定のデバイスを対象にしたものではありません。従って、それらを使用することは、デザインの移植性を妨げません (デザインで実際に必要になった場合のみ使用します)。統合ライブラリの検索、使用方法の詳細は、「[統合ライブラリの作成](#)」チュートリアルを参照して下さい。

それでは、Johnson Counter の回路図を作成してみます。

1. **Libraries** パネルのドロップダウンリストから FPGA Generic.IntLib を選択して下さい。
2. **Libraries** パネルで **SR8CLEDB** のコンポーネントを探します。ライブラリ名の下での **Masks** 編集ボックスで SR8CLEDB (または、名前の一部) と入力するかリストから探して **Libraries** パネルでブラウズすることができます。リストでコンポーネントを選択して **Place SR8CLEDB** ボタンをクリックするか、コンポーネント名を選択して回路図シートにドラッグ移動します。
3. カーソルにコンポーネントが表示されるはずですが、もし、カーソルにコンポーネントが表示されない場合は、回路図のワークスペース内にカーソルを移動して下さい。回路図上の適切な位置でクリックしてコンポーネントを配置します。コンポーネントのデジグネータの設定については、後でアノテートを実行するので心配ありません。



- この回路図では、8ビット入力バス(J8B_8S)、6つのインバーター、1つのORゲート(OR2N2S)、2つのフリップフロップ(FJKC)を使用します。図2のようにこれらのコンポーネントを配置します。
- NanoBoard にプラグインで接続するポートが必要になります。これらは、FPGA NanoBoard Port-Plugin.IntLib (**Libraries** パネルから利用できるデフォルトライブラリ) に用意されています。このライブラリから図2のように DIPSWITCH、TEST_BUTTON、CLOCK_REFERENCE、LED を配置します。

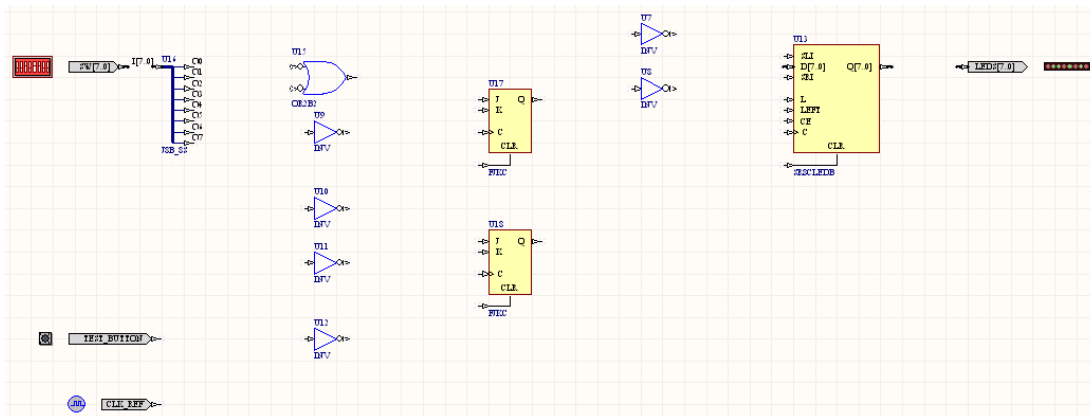


図2. 部品が配置された Johnson Counter 回路図

- 最後に、**Tools » Quiet Annotate** か **Tools » Force Annotate All** を使用してデザインにデジグネータを追加します。デジグネータは、この回路図上の全てのコンポーネントに自動で追加されます。

パワーポートの追加

2つの GND パワーポートを配置します。

- Place » Power Port** を選択するか、Wiring ツールバーの GND アイコンをクリックします。
- TAB** キーを押して、**Power Port** ダイアログを表示させます。Net が GND に設定されていることを確認して、Style のドロップダウンリストから **Bar** を選択します。**OK** をクリックし、GND ポートを配置します。
- 右クリックか **ESC** キーを押し、配置モードを解除します。
- SR8CLEDB のコンポーネントの D[7..]ピンにグラウンドバスパワーポートを追加する必要があります。Wiring ツールバーの **VCC Bus Power Port** ボタン  をクリックします。**TAB** キーを押して、**Power Port** ダイアログを表示させ、Net 名を GNDBUS[.]に変更し、Style が **Bar** に設定されていることを確認します。**OK** をクリックし、**Spacebar** を押してシンボルを回転させてから回路図に配置します。

オブジェクト間の接続

コンポーネントとポートを配置したので、次にオブジェクト間を接続します。オブジェクト間を接続するには明示的か暗黙的に接続するかの2つの方法があります。明示的な接続方法は、2つのオブジェクト間をワイヤで接続する物理的な接続です。暗黙的な接続方法は、ワイヤとネットラベルで接続

します。これは実際に物理的な接続がされておらず、2つのワイヤが同じネットラベルを共有していることで接続は確立されます。

この回路図では、ワイヤとバスが必要です。最初にワイヤを配置します。ワイヤとラインを混同しないように注意して下さい。ワイヤは電気的な接続に、ラインは図形などの描画に使用します。

1. ワイヤを配置する為に、**Place » Wire** [ショートカット P, W]を選択し、配置したい位置（通常は、コンポーネントのピンかポート）でクリックします。次に、接続したい位置にカーソルを移動し、もう一度クリックします。コンポーネントのピンかポートに接続するまでこの操作を続けます。右クリックか **ESC** キーを押してワイヤ配置モードを解除します。
2. ワイヤが交差するジャンクションに注意して、図3のようにワイヤを接続します。もし、2つのワイヤが交差しジャンクションが発生する場合、これらの2つのワイヤが接続されたことを意味します。もし、ジャンクションが無い場合は、接続されていません。ワイヤが接続される箇所では、自動でジャンクションが発生します。

他のワイヤ、コンポーネントのピン、ポートにワイヤを接続した時に接続が確立されます。もし、カーソルが、接続したいオブジェクトの上で赤い十字に変わる場合、ワイヤのノードをそこに配置すれば接続が確立されます。

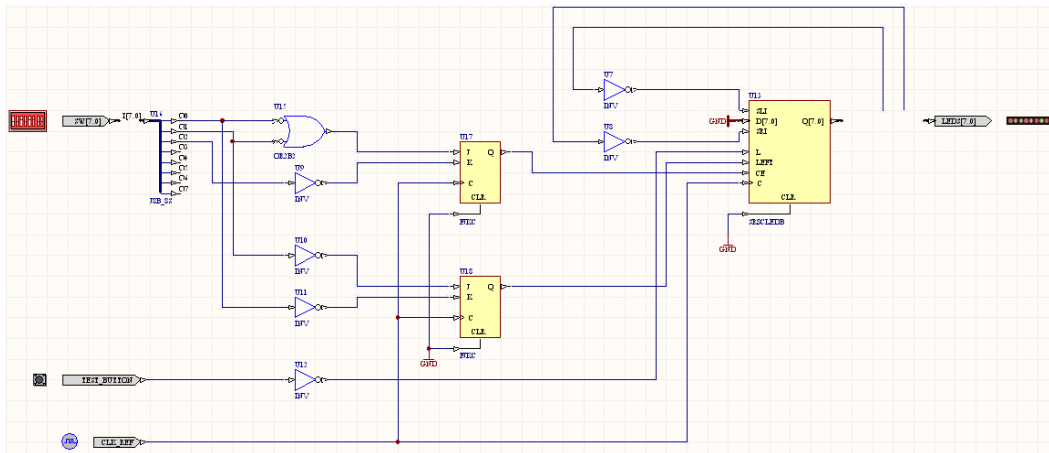


図3. ワイヤで接続された Johnson Counter の回路図

接続名の指定

上記で行った接続方法は、明示的な方法です。従って、ネットラベルは必要ありません。しかし、ネットラベルは回路図上の接続を分かりやすくし、問題を突き止めて接続箇所を容易に参照することができます。

1. **Place » Net Label** [ショートカット P, N]を選択します。カーソルのところにネットラベルが表示されます。
2. ネットラベルを配置する前に属性を編集する為、**TAB** キーを押して **Net Label** ダイアログを表示させます。**Net** の項目でネット名（例：LEFT）を入力し、**OK** をクリックします。
3. ネットラベルの左下（ホットスポット）がワイヤに触れるように、ネットラベルを配置します。ネットラベルがワイヤに触れるとカーソルは赤いクロスに変化します。

4. 下図のように他のネットにもネットラベルを配置します。ネットラベルは、各信号の識別ができれば、図4とまったく同じ名称を付ける必要はありません。右クリックか**ESC**キーを押し、ネットラベル配置モードを解除します。

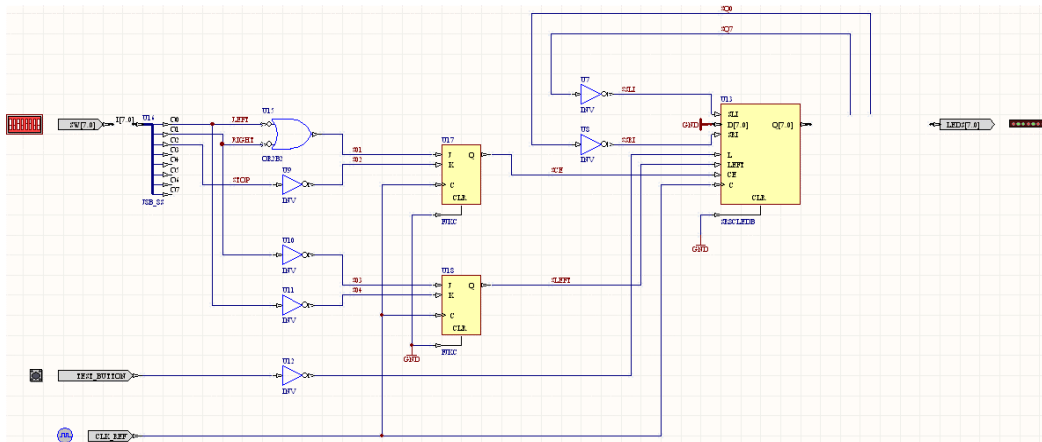


図4. ネットラベルが配置された Johnson Counter 回路図

バスの使用

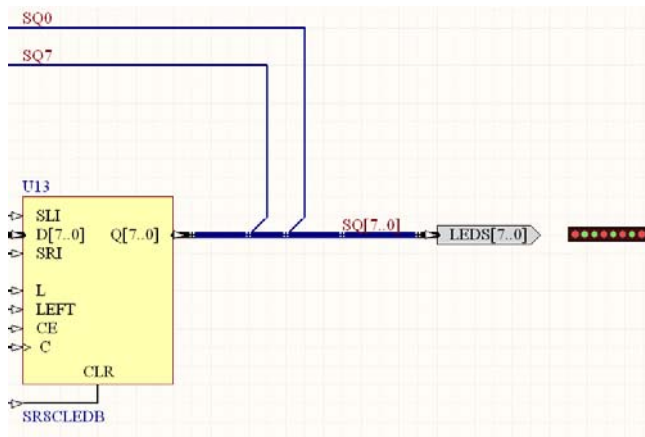
DXP 2004 では、FPGA 設計の為に複合のバスを使用することができます。バスは、単に信号がグループ化されたものではなく、どのようにバスの各信号が終了点に導かれるか指定するために使用します。

バスを使用する時、バスセグメントごとにネットラベルが常に必要であることに注意して下さい。また、バスから別のオブジェクトへ接続する場合は、常に左から右へ接続するようにし、双方のオブジェクトのバスサイズも同じである必要があります。

バスは、名前で判断されます。もし、ネットラベル名の最後が数値で、その前の名称が同じならば、ネットラベルをグループ化できます。

SR8CLED に LED ポートを接続する為に、図5のように **SQ[7..0]** というバスを作成します。

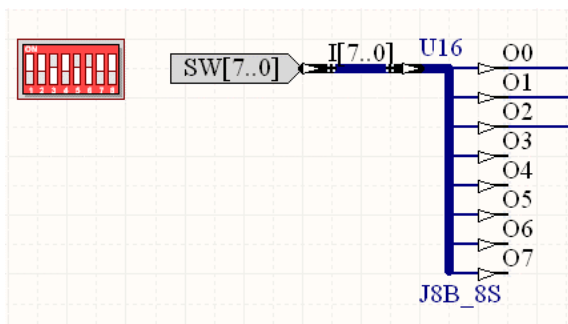
1. **Place » Bus** [ショートカット P, B] を選択して、ワイヤを配置する時と同じ配置方法でバスを配置します。



バスの接続をマッピングするとき、バスの範囲は左から右へと読み込まれます。

図5. バスとバスエントリの追加

2. バスに SQ[7..0] というネットラベルを配置します。バスには常にネットラベルを配置してください。確実に接続されている場合でも、ネットラベルがないバスは、バスの各要素がどのように終点に接続されているか明確に特定できない為、不定となります。
3. 次に、図5のようにバスエントリを追加します。Place » Bus Entry [ショートカット P, U]を選択し、SQ0 と SQ7 のワイヤから SQ[7..0] にバスエントリを配置します。バスエントリを配置している間に **SPACEBAR** を使用して回転させることができます。右クリックか **ESC** キーを押して配置モードを解除します。
4. DIP スイッチポートを J8B_8S に接続する為に、別のバスを追加します (図6 参照)。



ワイヤ上でバス形式のネットラベル(つまり“[]”)を使用するミスがよくあります。これは、バスだけで使用できる形式ですので、ワイヤ上に配置してもDXPでは動作しません。

図6. バスを備えた J8B_8S にDIP スイッチポートを接続

5. 回路図を保存し、プロジェクトを保存します。

デザインの確認

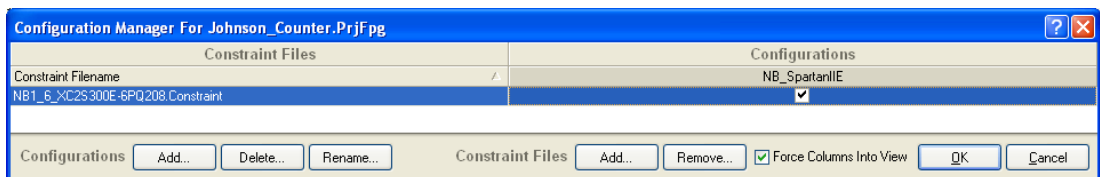
クロックデバイダのサブシートを作成する前に、プロジェクトのコンパイルを行い、*Options for FPGA Project* ダイアログ(**Project » Project Options**)の **Error Checking** タブ内で設定した電氣的、グラフィカルなチェックを元に回路図のチェックを行います。

1. **Project » Compile FPGA Project [project_name]**を選択します。Error または Fatal Error のメッセージは、自動で Messages パネルに表示されます。
2. Warnings も Messages パネルにリスト表示されますが、デザインウィンドウの下の **System** タブをクリックし、**Messages**(または、メニューから **View » Workspace Panels » System » Messages** を選択します)を選択してパネルを手動で表示させる必要があります。
3. **Compile Errors** ダイアログのエラーについての詳細を確認する為に、**Messages** パネルでエラーメッセージをダブルクリックします。エラー部分が回路図上でズームされハイライトされます。
4. エラー箇所を修正し、再度、プロジェクトのコンパイルを行います。回路図とプロジェクトファイルを保存します。

デザインのコンフィギュレーション

Johnson counter のデザインが完成しましたので、デザインで使用したい FPGA チップを指定する必要があります (例えば、NanoBoard ドーターボードの Xilinx Spartan IIE XC2S300E-6PQ208C チップ)。これを行う為に、**Configuration** と **Constraint** ファイルを追加します。**Constraint** ファイルは、NanoBoard の FPGA チップで使用するピンナンバーとデバイス名を決定します。

1. **Project » Configuration Manager** を選択し、**Configuration Manager for project** ダイアログを表示させます。ダイアログの Configurations の項目で **Add** ボタンをクリックし、**New Configuration Name** ダイアログで Configuration 名 (例: NB_SpartanIIE) を入力して **OK** をクリックします。Configuration 名は、確認し易くする為にターゲットに関連した名称にしてください。
2. Constraints の項目で **Add** ボタンをクリックして Configuration に Constraints ファイルを追加します (**Choose Constraint files to add to Project** ダイアログで NB1_6_XC2S300E-6PQ208.Constraint を選択します)。**Constraint** ファイルは、Altium2004\Library\FPGA フォルダに保存されています。**開く** をクリックします。
3. **Configuration Manager** ダイアログの Configuration のチェックボックスにチェックを入れ、**OK** をクリックします。



4. Settings という名前のフォルダが、プロジェクトに追加され、Constraints Files フォルダに **Constraint** ファイルが表示されます。
5. プロジェクトファイルを保存します。

Altera Cyclone デバイスをターゲットにするには

もし、Altera Cyclone チップをターゲットとする場合、NanoBoard に Altera Cyclone のドーターボードを注意して差し込みます。

1. FPGA プロジェクトに、例えば、Altera という名の新しい Configuration ファイル(**Project » Configuration Manager**)を追加します。

📖 コンフィギュレーションとコンストレイントについての詳細は、デザインのポータビリティ、コンフィギュレーション、コンストレイントのアーティクル、デザインを製造ボードに再ターゲットを参照して下さい。

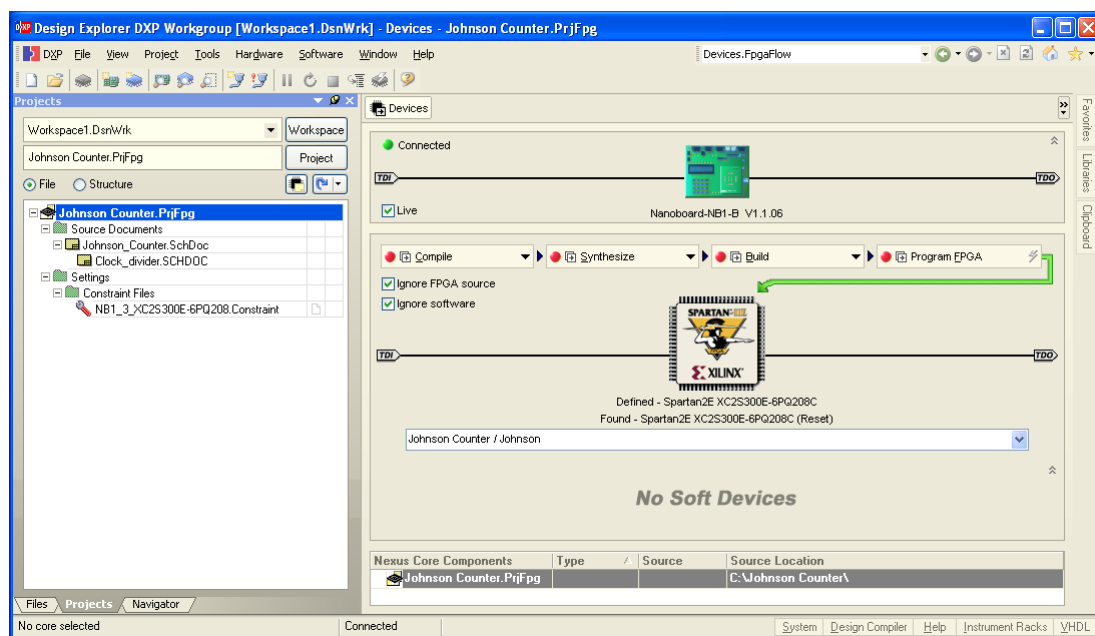
2. Altera Constraints ファイル (C:\Program Files\Altium2004\Library\Fpga\NB1_6_EP1C12Q240.Constraint) を追加します。Altera configuration が選択されていることを確認します。
3. “Devices 画面を使用してFPGA のプログラムを実行” の次の項目に進みます。

Johnson Counter のコンフィギュレーションが完了し、FPGA の準備ができました。残りの設計過程は、Devices 画面で行われます。

Devices ビューを使用して FPGA のプログラミング

Devices ビュー(**View » Devices**)では、(左から右への)ワークフローに沿って、FPGA にプログラムを送る為に必要な手順を示しています。この画面で以下のことができます。

- プロジェクトをコンパイル (エラーの確認)
- 論理合成 (EDIF ネットリストの作成)
- ビルド (例: EDIF ファイルの変換、デザインを FPGA に反映、FPGA の配置と配線、タイミング解析の実行、FPGA をプログラムする為に使用するビットファイルの作成)
- FPGA プログラム (ドーターボードの FPGA チップ (例: Xilinx Spartan IIE) にビットファイルをダウンロード)



このワークフローが完了すれば、NanoBoard 上の DIP スイッチをオン、オフすることでプログラムを実行できます。Johnson Counter デザインを FPGA へダウンロードするには：

1. NanoBoard が正しく接続されていることを確認し、スイッチをオンにします。Devices ビューで、**Live** ボタンをクリックし、**Connected** インジケーターが緑色に点灯していることを確認して下さい。

2. **Devices** ビューで、**Compile** をクリックします。編集が成功すると、赤色のインジケータは、緑色に変わります。もし、**Messages** パネルにエラーメッセージが表示された場合、回路図に戻りエラーを修正してからファイルを保存し、再コンパイルを行います。

3. **Synthesize** をクリックします。更新された **Synthesizing** ダイアログが表示されます。論理合成が成功すると、EDIF, VHDL ファイルと論理合成のログファイルが Generated [config_name] というフォルダに作成されます。この例で使用された **Configlation** の NB_SpartanIIE が、**Devices** ビューの **Spartan IIE** アイコンの下に表示されます。

論理合成中、ソースドキュメントは、中間のVHDL ファイルに置き換えられてからベンダーの配置・配線ツールに合った EDIF に論理合成を行います。論理合成中に発見されたエラーは、中間ファイルのエラーに基づくので、問題を解決する為には、ソースファイルに戻って下さい。**Messages** パネルにソースドキュメントと VHDL 中間ファイルのエラーが表示されますので、そのエラー内容をダブルクリックします。

ステージボタンの左にある矢印アイコンをクリックして、現在の段階までの全てのワークフローのステージを実行できます。例えば、**Program FPGA** ボタン上のこのアイコンをクリックすると、前の全てのステージを実行します。

4. **Build** をクリックします。これは、FPGA にダウンロードされる Bit ファイルを作成する為のプロセスまで進みます。それらのプロセスが完了するごとにボタンが緑色に変化することがわかります。必要なプロセスが全て完了した時に、**Build** ボタンが緑色に変化します(**Make PROM File** プロセスは、この例では必要ありません)。ベンダーに関するフィードバック情報を見る為、**Output** パネルを表示させます。

5. ドーターボードの **Spartan** チップにビットファイルをダウンロードする為、**Program FPGA** をクリックします。

6. **Program FPGA** プロセスが完了すると、**NanoBoard** 上の **DIP** スイッチを次のようにオン、オフ切り替えることによりプログラムを実行できます。

- スイッチ 1 で LED を右から左に表示させます
- スイッチ 2 で LED を左から右に表示させます
- スイッチ 3 でプログラムを終了させます

プログラムをリセットするには、LED の下にある **Test/Reset** ボタンを押して下さい。

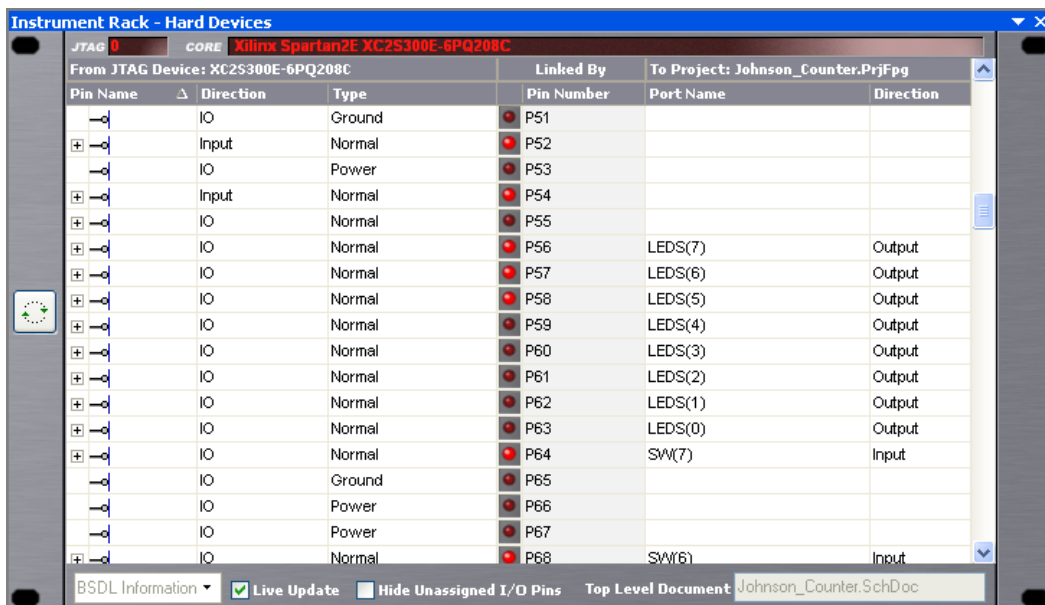
7. **Johnson counter** の本来の動作とは異なり、LED が同時に光っていることに注意して下さい。これは、クロックがデフォルトで 50MHz に設定されている為です。LED を連続して表示させるには、1/100 万までクロックを遅らせる必要があります。**Hard Devices** 装置を使用してプログラムが OK であることを確認した後に、回路図にクロックデバイダを追加します。

Hard Devices 装置を使用して LED をチェック

Instrument Rack の **Hard Devices** パネルでプログラムが正しく実行しているかチェックすることができます。この装置は、プログラムが実行している時に、ピン番号により LED を表示します。この表示は、**NanoBoard** のクロックに依存せず、LED を順に表示させる為に低い周波数は必要ではありません。

1. **Devices** 画面の **Hard Devices** の項目の **Spartan-IIE** アイコンをダブルクリックします。**Instrument Rack – Hard Devices** ダイアログが表示されます。



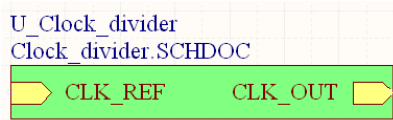


2. **Live Update** チェックボックスをクリックします。
3. スクロールダウンでピン番号 **P56** を表示させます。LED アイコンは、プログラムが実行されると同時に点灯します。

クロックデバイダのサブシートの追加

NanoBoard のクロックは、デフォルトでは 50MHz で動作しているので、LED 表示を遅らせる為に Johnson Counter の回路にクロックを 1/10 (CDIV10DC50 –デューティサイクル 50% バージョン) にする 6 つのクロックデバイダを追加する必要があります。FPGA のプログラムを行っている時点で、どのように階層設計が使用されるかを説明する為に、サブシートとしてクロックデバイダのサブ回路を作成します。

1. Johnson_Counter.SchDoc を開いて、クロックデバイダのサブシートを表すシートシンボルを配置します (図 7 参照)。**Place » Sheet Symbol** を選択します。配置する前に **TAB** キーを押して、**Sheet Symbol** ダイアログの **Properties** タブ内で designator 名 (例えば、U_Clock_divider) とファイル名 (例えば、Clock_divider.SchDoc) を入力します。**OK** をクリックしてダイアログを閉じてから、シートシンボルを配置します。
2. シートシンボルに、入出力の I/O タイプをそれぞれ設定した CLK_REF と CLK_OUT というシートエントリを追加します (**Place » Add Sheet Entry**)。



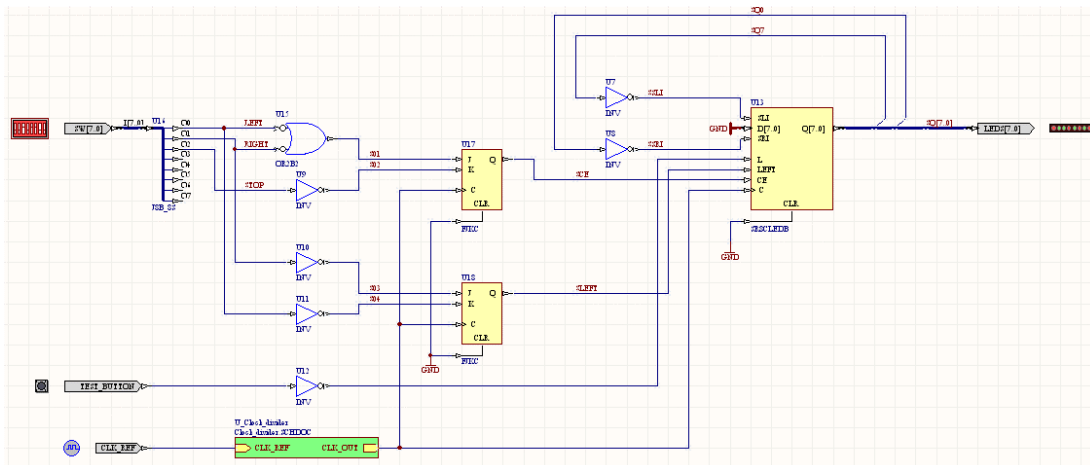


図7. Clock_divider のサブシートを表すシートシンボルが配置された Johnson_Counter の回路図

3. **Design » Create Sheet from Symbol** を選択し、サブシートを作成します。新規のシートシンボルを配置する位置にカーソルを移動し、クリックします。シートエントリーの入力/出力の方向を入れ換ええない場合は、**Confirm** ダイアログで **No** をクリックします。新しい回路図が作成され、自動で追加された CLK_REF と CLK_OUT のポートが配置された回路図が開きます。
4. 図8の様に、FPGA Generic.IntLib から CDIV10DC50 という部品を選択し、クロックデバイダの部品を6つ配置します。designator の設定について心配することはありません。部品を配置した後に、**Tools » Annotate Quiet** を選択すると、自動で Designator が設定されます。

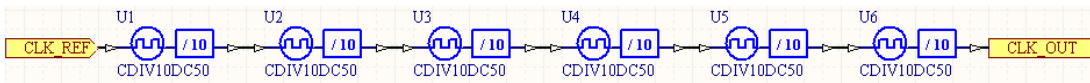
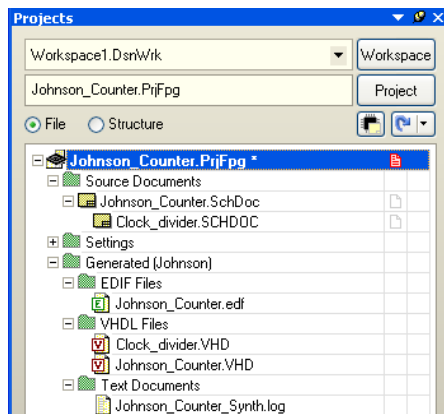


図8. 部品とポートが配置された Clock_divider.SchDoc

5. 回路図とプロジェクトファイルを保存します。
6. エラーのチェックを行う為に、デザインを再コンパイルします。エラーがある場合は、修正し保存します。
7. コンパイル後、**Projects** パネルでプロジェクトのシート階層を確認できます（サブシート (Clock_divider)は、Johnson_Counter の回路図の子シートとしてプロジェクトで認識されます）。



8. LED 表示の点灯速度が右から左へ (DIP スイッチ 1) または、左から右へ (DIP スイッチ 2) 遅くなった状態を確認する為に、**Devices** 画面に戻り、FPGA のプログラムを再度、行います。

クロックデバイダへ VHDL ファイルを追加

FPGA プロジェクト内のクロックデバイダ回路図サブシートに VHDL ファイルを代用します。この VHDL ファイルは、1/100 万までクロック率を下げます。VHDL ファイルは、シートシンボルを介して回路図にリンクされます。

1. **Projects** パネルで FPGA プロジェクト名を右クリックして **Add Existing to Project** を選択し、プロジェクトに VHDL ファイルを追加します。**Choose Documents to Add to Project** ダイアログから VHDL ファイル (Clock_divider.VHD) を選択します。このファイルは、Altium2004\Examples\Tutorials\Getting started with FPGA Design フォルダに保存されています。

もし、ゼロから VHDL ファイルを作成する場合、FPGA プロジェクト名を右クリックし、**Add New to Project » VHDL Document** を選択します。そして、図 9 のようなコードを入力し、ドキュメントを保存します。

```

Clock_divider.VHD

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

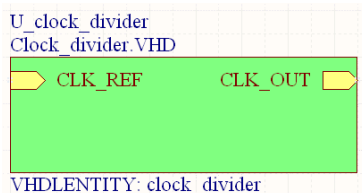
entity clock_divider is
  port (
    CLK_REF : in std_logic;
    CLK_OUT : out std_logic
  );
end entity;

architecture RTL of clock_divider is
begin
  process(CLK_REF)
    variable i : integer range 0 to 999999;
  begin
    if rising_edge(CLK_REF) then
      if i = 0 then
        CLK_OUT <= '1';
        i := 999999;
      else
        CLK_OUT <= '0';
        i := i - 1;
      end if;
    end if;
  end process;
end architecture;

```

図9. Clock_divider.VHD.

2. 次に、回路図サブシートである Clock_divider.schdoc のシートシンボルを交換する為に、新しい VHDL ファイルからシートシンボルを作成します。Johnson_Counter.SchDoc の回路図が開いている状態で、**Design » Create Sheet Symbol from Sheet** を選択して、新しいシートシンボルを作成します。**Choose Document to Place** ダイアログから Clock_divider.VHD を選択し、**OK** をクリックします。シートシンボルがカーソルに表示されます。**TAB** キーを押して、**Sheet Symbol** ダイアログを表示させます。VHDL Entity パラメーターが追加されたことを確認する為に、**Parameters** タブをクリックします。**Visible** オプションが選択されていることを確認してから **OK** をクリックします。Johnson_Counter の回路図にこのシートシンボルを配置する為に、Clock_divider.schdoc のシートシンボルの下でクリックします。



もし、VHDL ファイルが複数のエンティティを含んでいる場合、VHDLENTITY パラメータで、どのエンティティかを示します。

- Clock_divider.schdoc のシートシンボルを削除し、その場所に Clock_divider.VHD の新しいシートシンボルを移動します。ワイヤが正しく接続されているか確認して下さい。

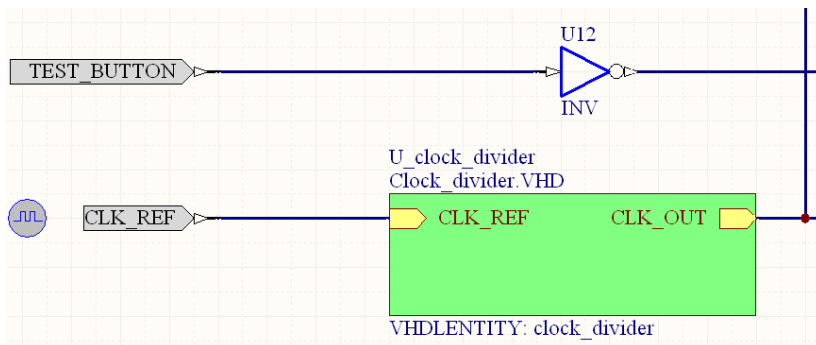


図 10. 回路図に配置された Clock_divider.VHD のシートシンボル

- 回路図を保存します。
- Projects パネル内のドキュメント名 (Clock_divider.schdoc) を右クリックし、**Remove from Project** を選択し、FPGA プロジェクトから Clock_divider.schdoc ファイルを削除します。この回路図は VHDL ファイルが追加された為、不要です。FPGA プロジェクトファイルを保存します。
- 最後に、VHDL ファイルでの遅延によりクロックが遅くなっているのを確認する為に、**Devices** ビュー (**View » Devices**) を表示します。そして、FPGA チップの編集、論理合成、ビルド、プログラムの全てを行う為に、**Program FPGA** をクリックします。もし、FPGA のプログラムが成功すれば、以前のように **Johnson Counter** を実行することができるはずです。

更新履歴

Date	Version No.	Revision
16-Jan-2004	1.0	New product release
23-Sep-2004	1.1	Clock divider components updated

Software, hardware, documentation and related materials:

Copyright © 2004 Altium Limited.

All rights reserved. You are permitted to print this document provided that (1) the use of such is for personal use only and will not be copied or posted on any network computer or broadcast in any media, and (2) no modifications of the document is made. 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, CAMtastic, Design Explorer, DXP, LiveDesign, NanoBoard, NanoTalk, Nexar, nVisage, P-CAD, Protel, CircuitStudio, Situs, TASKING, and Topological Autorouting and their respective logos are trademarks or registered trademarks of Altium Limited or its subsidiaries. All other registered or unregistered trademarks referenced herein are the property of their respective owners and no trademark rights to the same are claimed.