

マルチチャンネルデザイン の概念

アーティクル



nVisageDXP
PretelDXP

True multi-channel designs can be created in DXP,
with single or multiple levels of repeats

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.

マルチチャンネルデザイン の概念

概要	1
マルチチャンネルデザイン の概念	1
マルチチャンネル接続	2
チャンネルの指定	3
PCB デシグネーターのためのルール	4

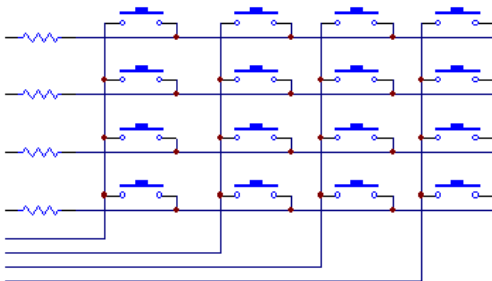
概要

このアーティクルでは DXP が装備しているマルチチャンネルの設計機能を紹介し、チャンネルごとに共通あるいは分配させるべきネットの取り扱い方や、PCB での各チャンネルのクラスとルールについて解説しています。

マルチチャンネルデザイン の概念

DXP では、チャンネル内に他のチャンネルをネストして配置することができるマルチチャンネルデザインを装備しています。

デザインには同じ回路の繰り返しが含まれる場合が多くあります。例えばある基板では同一回路を 32 回も繰り返し使っていたり、あるいは 8 つの同一回路のサブチャンネルをそれぞれに持つ 4 バンクで構成している場合があります。従来の回路図入力ではこういった設計を、PCB レイアウトと関連付けるために“フラット階層”設計で対処せざるを得ず苦労してきました。回路図入力時の初期段階では、回路図のコピーアンドペーストで比較的容易に作業できますが、回路図プロジェクトに一旦修正が発生すると、今まで作業した回路図の書き直しは困難でした。DXP はこういった繰り返しの多いプロジェクトに効果的な真のマルチチャンネルデザインの機能を提供しています。ただ一枚の回路図を修正してプロジェクトを再コンパイルするだけで、全チャンネルに渡って必要な変更を反映させることができます。また DXP はマルチチャンネルだけでなく、回路図のネストもサポートしています。下図のような 16 個のキーパッドを持つ回路図を例に挙げて説明しましょう。



DXP の回路図エディタでこんな簡単な回路図なら、わずか数分で書き上げることができます。しかし後に、キーパッドのシンボルやパラメータを修正する必要がある場合や、現状の 4x4 を 10x12 に拡張したい場合にはどうでしょう？

グループ編集ツールの助けを借りたとしても、この修正はひどく退屈でエラーも生じやすい作業になります。そこで回路図の 2 枚のシートに 2 種のコンポ

一ネット（抵抗とスイッチ）だけをそれぞれ配置したと仮定してみます。上階層のシートシンボルに Repeat パラメータを記述してあれば、DXP のコンパイラは 1 枚のシートを参照し仮想の複製（チャンネル）を指定した個数分だけ作成します。まず抵抗を配置した回路図でスイッチを 4 個並べるためにスイッチを配置したシートを 4 回リピートし 1 行に相当する部分を作成します。さらにトップ階層で抵抗 1 個とスイッチ 4 個並んだこの回路図を 4 回リピートして回路全体を形成させます。以上の操作でマルチチャンネルが複数のチャンネル内に収められ、それに応じた接続のマトリクスが確立されています。

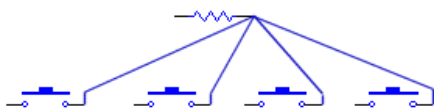
このマルチチャンネルの機能により設計者の作業が容易になり、上の例ではスイッチや抵抗に対しての変更を何度も行う必要は無く、一回の修正で行うことができるようになります。また、スイッチ配列のマトリクスを拡張したい場合は、シートシンボル内の Repeat 記述を単に修正するだけの作業で行えるようになります。

マルチチャンネルデザインを素早く作成するには、DXP でのプロジェクト内での接続方法に慣れている必要があります。

マルチチャンネル接続

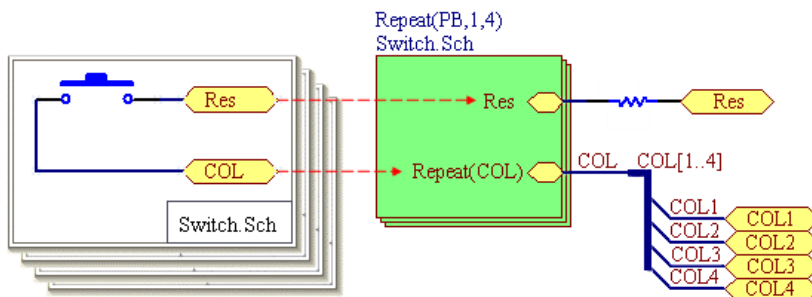
他のマルチシートデザインの場合と違いマルチチャンネルデザインでは Net Identifier scope の設定を Hierarchical (sheet symbol <-> port) にしておく必要があります。Project Options ダイアログでこの設定を Automatic にしておくのが最良ですが、繰り返されるシートに信号を伝達させる唯一の方法は、親シートのシートエントリに対応した子シートのポートへの接続であると理解してください。この制限はシートエントリがポートやネットラベルと異なり、Repeat 記述で作成される繰り返されたチャンネルを扱うように作られているからです。

マルチチャンネル設計で利用できるシート間の接続には、全てのチャンネルに共通したネットと各々に固有なネットの 2 種類があります。これらのタイプを上記のキーパッドの例で説明すると、各スイッチには抵抗に直接接続させるピンがありました。これは各チャンネルの同じノードが親シートの 1 つのノードに対し互いに結ばれる共通のネットの下図のように接続されています。



この方法は DXP のマルチチャンネルデザインで一番簡単なネット接続の方法で、シートエントリとチャンネルデザイン内の一致するポートが接続されます。

スイッチのもう片側は縦方向のスイッチ同士を結んでいます。重複したチャンネルから接続情報の伝達のために、Repeat 記述をシートエントリに付加しています。



この図は、2 種類の接続タイプを表しています。

2 つのシートエントリに違いがあることに注意して下さい。シートエントリ Res は、同じ名前のポート同士をそれぞれ接続します。シートエントリ COL

にはポート名に **Repeat** コマンドが付加されています。これはネットラベルが付けられたワイヤやバスに対し、コピーされた 4 枚のシートから 4 本のネットが引き出されていることを意味しています。このポイントから必要なノードを分離して接続したり、さらにポートを通して別の階層へ接続することができます。

一旦 **DXP** の接続構造に従ったマルチチャンネルプロジェクトを作成すれば、各チャンネルのコンポーネント名称の割り当て方が理解できるでしょう。

チャンネルの指定

チャンネルとは仮想的なシートであり **DXP** のシンクロナイザーは各チャンネルに他と区別したルームを割り当てます。しかしその前にコンパイラーで、全チャンネル内の全コンポーネントに対して固有な名称を割り付けておく必要があります。ネスト化したチャンネルを使っている場合には、これらの名称は、**Project Options** ダイアログの **Multi-Channel** タブで設定した内容に従って割り当てられます。

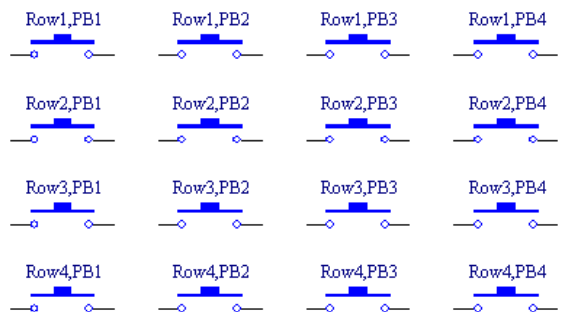
基本的には 2 種類の設定方法があります。1 つ目はルームとコンポーネントの名称をフラットまたは階層的に割り当てるかです（もしデザインがネストした階層を含む場合この方法が重要になります）。2 つ目はインデックスに数字またはアルファベットのサフィックスを付加するかです（両方を考慮するオプション設定もありますが、ネストしたチャンネルの名称を区別しやすくします）。

コンポーネントデザイナーに指定するフォーマットには、特定のチャンネルとルーム名を含んでいます。必ず必要というわけではありません。通常のデザイナーフォーマットでは、コンポーネントとチャンネルの両方のプリフィックスとサフィックスが含まれます。7 つの予約されたキーワードのうちの 2 つ (**\$Component** と **\$RoomName**) は、これらを自動で行います。

\$Component は、**\$ComponentPrefix** **\$ComponentIndex** の組み合わせになっています。例えばコンポーネントパート **U3A** は、プリフィックス (**U**) とインデックス (**3A**) から成り立っています。同様に **\$RoomName** は、チャンネルプリフィックスとサフィックスの組み合わせで、その正式な形は選択したスタイルと、指定したデザインおよびルームにより変化します。もしデザインがネスト化したチャンネルを含まないのであれば、階層的部分のスタイルは無関係に、数字またはアルファベットの部分が **\$RoomName** で示されたチャンネル表記に変化します。しかしデザインにネスト化したチャンネルがあり、スタイルを指定する階層構造のルームを使用している場合は、**\$RoomName** を **\$ChannelPrefix**・**\$ChannelAlpha**・**\$ChannelIndex** に置き換えることはできません。これは、**Repeat** 記述で使用したチャンネル名から、マルチレベルプリフィックスを構築する為です。

例えば先に解説したキーボードのプロジェクトで説明すると、**Room Naming Style** を **Numeric Name Path** に設定した場合、**\$RoomName** の記述は **Level Separator** と **Designator Format** がコンマで区切られ、デザインをコンパイルした後の 16 個のボタンは図のように記述されます。

Alpha Name Path を選択した場合、デザイナーは、**RowA,PBA** や **RowA,PBB** などになります。**Mixed Name Path** を選択していた場合なら、**RowA,PB1** や **RowA,PB2** などとなります。しかしもしフラット構成を選択していた場合には、**Row** チャンネルは参照されずデザイナー



一が PB1 から始まり PB16 で終わるか、PBA から始まり PBP で終了するような名称が割り当てられます（数字またはアルファベットのサフィックス）。これらは **Designator Format** フィールドで指定したコンポーネント自体への参照を除外するための機能ではなく、各チャンネルにコンポーネントが含まれる場合にのみ実行されます。

これらは物理的なデシグネーターよりむしろ論理的に表すためのオプションなので、余りに字数が多くなりすぎて PCB に適合しなくなることを憂慮する必要は無く、デシグネーターに希望する情報が含まれていることを確認してください。

PCB デシグネーターのためのルーム

マルチチャンネルデザインは回路図時だけでなく、後の行程でも便利になります。PCB 設計時の利点は、ボード外形の近くに自動で置かれたルーム内に、それぞれのチャンネルのコンポーネントが自動的に展開されます。ルームに基づいた部品配置のツールは、マルチチャンネルデザインでのレイアウト作業に於いて強力な手助けとなるでしょう。

DXP の **Situs** オートルーターは、ルーム内に含まれた接続を配線できる機能を備えています。また逆に、**Unrouting** ツールも同様に、ルーム内の配線を削除する機能を備えています。各チャンネルに対応するコンポーネントは同じ **Channel Offset** の値が割り当てられており、セレクトまたは編集する際には一括して実行することが可能です。

さらに特徴的な機能として、あるルームから他のルームへとフォーマットをコピーすることができることです。この機能を使えば 1 つのルームに行った変更を即座に別のルームへ反映させることができます。コピー元とコピー先のルームが同じチャンネルクラス(同じシートから複製されたことを意味します)であるならば、更新は一度に全ての同一クラスメンバー全体に反映できます。フォーマットをコピーする内容は、ルームのサイズや形状の変更・フットプリントの配置や配線の削除範囲まで制限を設定することができます。

PCB エディタには、マルチチャンネルデザインを最大限に活用できるように多くのツールが追加されました。一例をあげるなら、ボードの反対側にルームごとオブジェクトを移動したい場合は、**Design » Rooms** のサブメニュー **Move** コマンドで移動中に L キーを押すと、配置と配線を含むルーム全体が層を変えて移動したことが確認できるでしょう。

マルチチャンネルデザインの根本的に有利な点、それが回路設計時だけでなくボード設計時の省力化にも寄与できる点です。もう同じ作業を 32 回も繰り返して行う必要はありません。

マルチチャンネルデザインの利点についての詳しい実例は、**Altium\Examples\Peak Detector** フォルダの PCB プロジェクトを参照して下さい。