

第7章

パラメタライズ設計と不要な出力ラッチ生成の解消

VHDLの記述スタイルと効率

この章では、VHDLの記述効率を改善する便利な手法と、よく陥るトラブルを回避するヒントを紹介します。

VHDLで回路を記述していると、よく似た回路を何度も記述していることがあります。回路図ベースの設計でも、こうしたことはよくあったものですが、やはり何とかしたいところです。そこで、プログラミング言語の便利な点を取り込んだ機能が、VHDLには用意されています。これまでも何度か触れてきた「パラメタライズ設計」です。

この手法は、コンポーネントやプロシージャ、ファンクションなどとの併用はもちろん、通常の回路記述でも役に立ちます。特に、バス幅の異なる

設計や段数の異なるシフトレジスタなどを、パラメータの操作だけで共通設計化できるため、一度記述したVHDLコードの再利用が容易になります。

procedureやfunctionを作成する際には、通常、一部の要素をパラメータによって可変にしておきます。こうすれば、procedureやfunctionを特定の回路専用のものとしてではなく、別の回路でも利用できる汎用性をもったものにすることも可能です。

このような特徴をもつパラメタライズ設計は、従来の回路図ベースの設計にはない、VHDLならではの柔軟性と言えるでしょう。

7-1

コンポーネントをパラメータで制御する パラメタライズ設計の基本

パラメタライズ設計は、特定の構文を記述すればできあがり、というわけではありません。VHDLによって回路を記述する段階からパラメタライズを意識した構成と、パラメータの記述のコンビネーションによって初めて実現します。

よく使われるgeneric文やgenerate文だけでなく、ほかにもさまざまな要素を駆使してパラメタライズを行います。もちろん、generic文、generate文などは、パラメタライズの中心的な役割を担う構文です。generate文については第6章の最後ですでに紹介していますので、ここではgeneric文を使ったメモリ回路の事例を紹介しましょう。

● generic文

generic文はエンティティ部のport宣言の前に記述します。これによって、**ポートの記述にパラメータを利用できる**ようになります。このパラメータの記述により、設定を外部から受け取れるようになります。

また、component宣言文の中に記述することもできます。この場合、対応するcomponentインスタンスのport map宣言の前にgeneric map宣言を記述し、パラメータの対応付けを行います。これによって、エンティティ部でしか有効化できないパラメータ記述を、アーキテクチャ部にも導入することができるようになります(com-

図1 generic文の構文

```
generic ( <信号宣言文> [ := <デフォルト初期値> ] ;  
         <信号宣言文> [ := <デフォルト初期値> ] ;  
         ⋮  
         <信号宣言文> [ := <デフォルト初期値> ] );
```

第8章

統合開発ツールを使った開発の実際

VHDL による FPGA への実装

この章では、VHDLによる回路設計例として、ザイリンクス(Xilinx)社の“Spartan 3 Starter kit”を使った、1秒ごとにカウントアップする7セグメントLED表示回路の試作例を紹介します。

試作する回路には、簡単なクロックの取り込みのしくみや、I/Oバッファ、10進カウンタ、7セグメントLEDのダイナミック点灯といった回路を組み込んでいます。この段階では特に実用性は

ありませんが、今後の設計に利用できそうなサンプル記述の詰め合わせです。

また、外部の水晶発振回路からの入力やプッシュボタン・スイッチの入力、クロック信号の出力などに、3.3V CMOSのバッファ回路を追加しています。これは、ザイリンクス社が提供するライブラリに含まれるプリミティブを、直接にコンポーネント宣言でアクセスして利用しています。

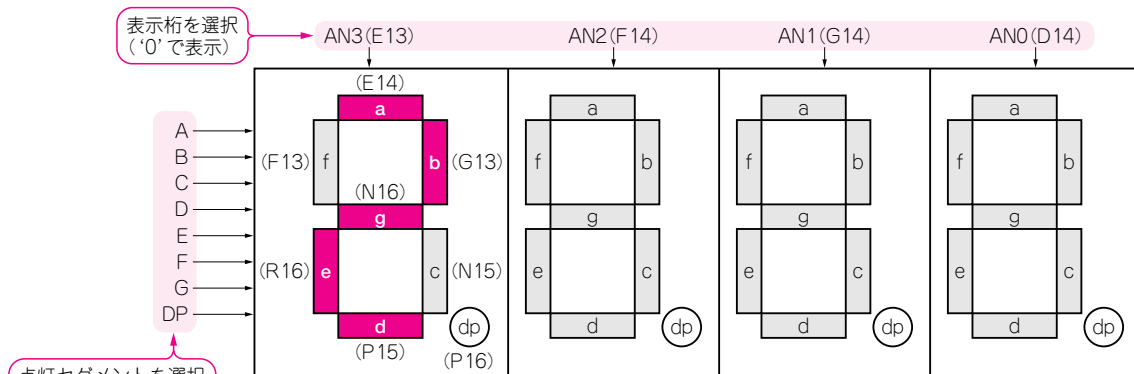
8.1 「Spartan 3」の評価キットを使用する ターゲット・デバイスとLED表示器の仕様

● ターゲット・デバイス

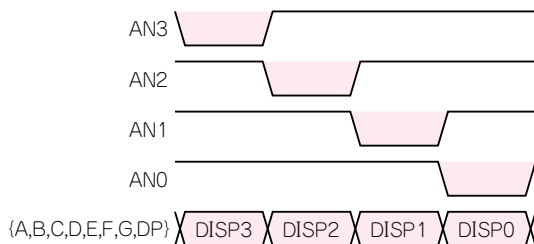
今回の1秒カウントLED表示器を実際に実装

したターゲットは、ザイリンクス社のSpartan 3 (XC3S200 FT256)を搭載したFPGA評価キット

図2(2) 評価用ボードの7セグメントLED



(a) 各桁と各セグメントの配置。()内はFPGAのピン番号



(b) 表示のタイムチャート

ここでは、プログラマブル・デバイスの基本的な事柄を紹介します。何しろ進歩の速い業界ですから、次々と新しい技術、新しいデバイス、新しい製品が登場してきます。そのすべてのフォローは容易ではないため、ここで紹介していないことや最新の情報は、デバイス・ベンダや取り扱い各社から入手してください。

CPLDの特徴

CPLDは、Complex Programmable Logic Deviceの略で、「複雑なPLD」という意味です。PLDの技術を基本に、それを大規模化したものがCPLDと考えてよいでしょう。

そのPLDの基本的な内部構

成は、いくつかのファンクション・ブロックと、そこに組み込まれた複数のマクロセルと呼ばれるものから成っています。これをいくつも搭載し、相互配線とそれに付随する回路を追加したものがCPLDです。

図B-1はファンクション・ブロックと、ファンクション・ブロックの内側の概念図です。

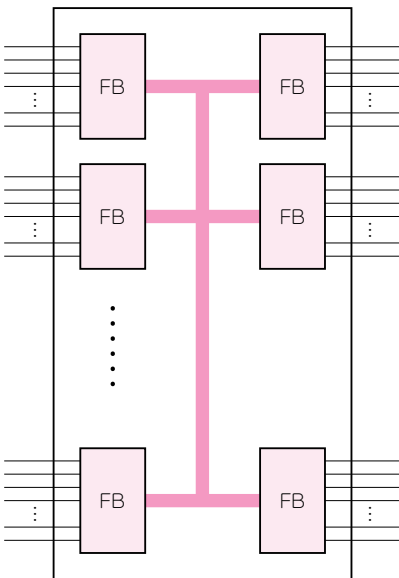
おおざっぱに言えば、CPLDはファンクション・ブロックやI/Oブロック(外部入出力回路ブロック)をいくつも搭載し、それらを相互接続するクロスバ・スイッチのようなスイッチ・マトリクスをもっています。こうした構造によって、すべてのファンクション・ブロックとI/Oブロックを相互に接続することが可能です。

● マクロセル

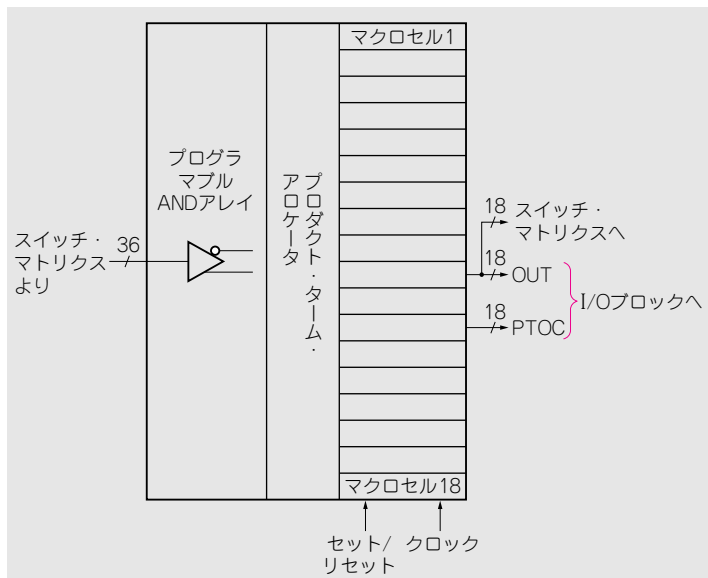
基本的に一つのファンクション・ブロックには、複数のマクロセルが組み込まれており、マクロセルは、一つのフリップフロップ(レジスタとも呼ばれる)と、いくつかの組み合わせ論理回路がセットになったものです。マクロセルの等価回路の例を図B-2に示します。

このマクロセルに付属する組み合わせ論理回路は、コンパイラなどによる論理の簡約化を前提とした論理積ANDと論理和OR、そしてNOTだけで構成されており、プロダクト・ターム(共通項、積項)と呼ばれます。ちなみにプロダクトは、論理数学における「共通、AND」の意味です。

図B-1 CPLDのファンクション・ブロックの構成



(a) CPLDの内部



(b) ファンクション・ブロック (FB)