

第6章 回路図入力で作ったカウンタを 言語で作り直す

HDLを使った回路設計にTRY!

大中 邦彦
Kunihiko Ohnaka

● 回路図エディタで大きな回路を書くのはたいへん
第2部では、回路図入力方式で7セグメントLEDの点灯回路を設計しました。

回路図は視覚的に動作を捉えるのにはもってこいの入力方法です。回路図に慣れてくると「こういう接続の形をしている回路は、**な動作をするんだ」というふうに直感的にわかるようになります。

第5章で、実際に回路図エディタを操作してみた方は、どんな印象をもったのでしょうか? 「お絵描きツールみたいで面白かった」「マウスだけでCPLDの回路が設計できるとは思わなかった」といった好印象だった人もいるかもしれませんが、「繋がらないといけない線が多くて面倒」「余計な線どうしが繋がったりして、操作が難しかった」「画面に回路全体を収めるとパーツが小さくなって見にくい」など、マイナスの印象をもった人もいるでしょう。

そうなんです。実は回路図入力は、取っ付きやすさはあるのですが、少しでも大きな回路を作ろうとすると意外にめんどろな方法なのです。

そこで、現場のデジタル回路設計技術者は、大きなデジタル回路を設計するために「ハードウェア記述言語」と呼ぶデジタル回路開発用の専用言語を使っています。マウスではなく、キーボードを使って回路を設計しています。

デジタル回路を記述できる 言語のいろいろ

● VHDLとVerilog-HDLが2大メジャー

ハードウェア記述言語は、英語ではHardware Description Languageと書き、一般に「HDL」と呼んでいます。書店の専門書コーナーに行くと、タイトルにHDLと書かれた本を何冊も見つけることができます。そこで本章でも、以降はハードウェア記述言語のことをHDLと省略して書くことにします。

HDLは、特定の一つの言語をさすものではなく種類があります。現在の代表的なHDLには図1に示すような種類があります。

AHDLやABELはかなり古い言語で、今ではあま

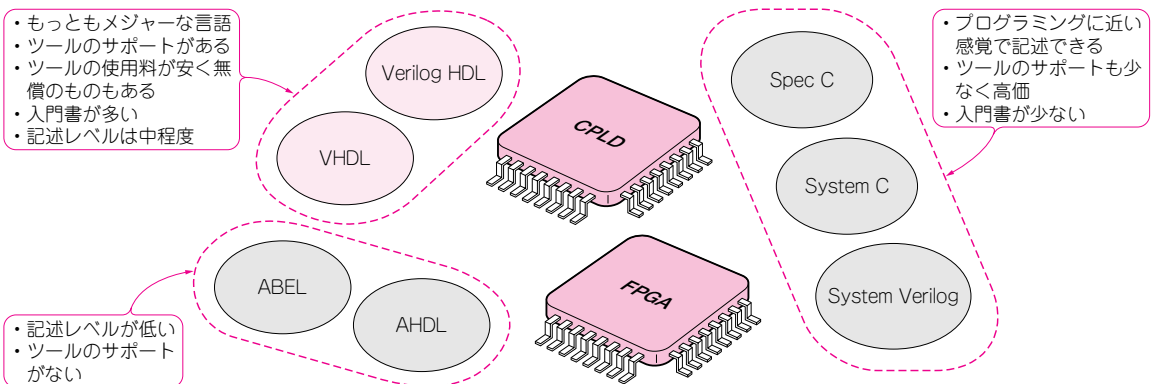


図1 ハードウェア記述言語のいろいろとその特徴

Keywords

AHDL, ABEL, VHDL, Verilog HDL, SpecC, SystemC, System Verilog, コンパイラ, コンフィグレーション, Quartus II, 74160, 7447, BCD, エンティティ, rtl, PROCESS, ARCHITECTURE

り使われていません。

VHDLとVerilog HDLはもっともメジャーです。どちらも一長一短で大差がないため、どちらを使うかは設計者の好みや使い始めたときの条件によるようです。

SpecC, SystemC, System Verilogなどは、VHDLやVerilog HDLよりもさらに自由に回路を書けるようにくふうされた言語です。まだ広く普及はしていませんが、今後メジャーになっていく可能性があります。

● 専用のコンパイラが必要

文字で入力されたハードウェア記述は、専用のコンパイラによってコンフィグレーション・データ(CPLDに書き込める形式の結線情報データ)に変換されます。この変換の際、**使用している言語専用のコンパイラが必要になります**。付録CPLD基板の場合は、アルテラのMAX IIに対応したコンパイラが必要です。

コンパイラは無償のものもありますが、多くは有償です。安価でないものも多いですから、一式そろえるには財布と相談しなければなりません。お金を出す覚悟があっても、自分の使いたいCPLDに対応したコンパイラが存在しないことさえあります。

このように考えていくと、現時点で手軽に使えるHDLはある程度絞られます。それは、先ほどメジャーなものとして紹介した**VHDLとVerilog HDL**です。

この二つが広く使われている理由として、**コンパイラなどの開発ツールが充実している点**があげられます。実際、現在市場で手に入るプログラマブル・デバイスのなかで、VHDLやVerilog HDLが使えないものはほとんどないと言える状況です。Quartus II評価版のように無償で利用できるコンパイラも多くあります。VHDLやVerilog HDLのコンパイラが無償で利用できるというのは10年前には考えられなかったことです。現在は、手軽にデジタル回路設計を始められる時代です。

● 本特集ではVHDLを使う

特別な理由はないのですが、本特集ではVHDLを使用して、言語を使ったデジタル回路設計の方法を解説します。Verilog HDLを使ってみたい人は、ほかの文献を参考にしてください。

VHDLとVerilog HDLでは確かに記述方法が少し異なるのですが、**できることにはそう大きな差はないので、VHDLに慣れておけば、将来Verilog HDLを使うときにも役に立つはず**です。

● 回路図の代わりにVHDLで回路を書く

第5章では、2個のプッシュ・スイッチが押された回数を2個の7セグメントLEDに表示する簡単な回路(図4)を作りました。そのときは回路図入力方式でMAX

IIの内部回路を設計しましたが、ここでは、その回路とほとんど同じものをVHDLを使って作ってみます。

両者を比較してみることで、言語(VHDL)を使った回路設計の世界を実感できると思います。

第5章で作ったカウンタの構成部品をVHDLで作り直す

■ 第5章のおさらい

第5章 図4の回路のおさらいをしましょう。

この回路では、プッシュ・スイッチの入力を74160のCLK端子に接続しました。74160は、0から9までの数を数えられるカウンタです(p.176参照)。74160はいろいろな使いかたができますが、第5章ではCLK端子にパルスを入力して、そのパルス数を数えるようにしました。

プッシュ・スイッチを押すたびにパルスが生成されるので、プッシュ・スイッチが押された回数を0から9まで数えられるというわけです。

数えた結果は、4ビットの2進数の値として7447という7セグメントLEDデコーダ回路に入力されます。7セグメントLEDデコーダとは、入力された2進数の値をLEDに表示する回路です。2進数で4ビットだと、扱える数値は0から15までの16とおりです。しかし、カウンタは0から9までしか出力しないので、10～15については気にしないことにします。

また、74160と7447を組み合わせて作ったカウンタをButtonCountという名前のエンティティにし、さらにその回路を二つ並べて、2個のボタンが押された回数を数えられるようにしました。そして全体をButtonCount_topというエンティティ(トップ・レベル・エンティティ)にまとめてMAXIIに収めました。

■ 74160と7447もVHDL化する

● 第5章で使用した74160と7447の違い

図2に示すのは、これから作る回路の全体図です。第5章 図8と見比べるとわかるように、両者を対比しやすいうように、ほとんど同じ構造にしました。違いは、第5章 図8で**74160となっていた箱(エンティティ)をBCDCounterというエンティティに置き換え、7447の部分をDecode7Segmentというエンティティに置き換えている点**です。

ではさっそく、BCDCounterとDecode7Segmentの二つのエンティティをVHDLを使って作ってみたいと思います。

● 74160のVHDL版「BCDCounterエンティティ」の機能

図3に示すのはBCDCounterの仕様です。