

# VHDL/Verilog HDL の基本プログラム集



第5章

第3章と第4章において開発ツールの使い方とシミュレーションの手順を習得した。次に，ハードウェア・プログラミングで使用されるいろいろな基本プログラムを見ながら，その動作をシミュレーションしてみよう。さまざまなプログラムをシミュレーションすることで，VHDLとVerilog HDLのプログラミングのイメージをつかもう！ (編集部)

三好 健文，高前田 伸也

新しいプログラミング言語の習得には，さまざまなサンプル・プログラムを見て実際に動かしたり，コピー&ペーストで自分のプログラムに移植してみたりするのが早道です。本章では，HDLコードで記述する基本的なハードウェア・モジュールとして，カウンタ，合計値計算モジュール，7セグメントLEDの点灯制御，乱数生成器，シリアル通信モジュールをVHDLとVerilog HDLの両方の記述例を紹介します。いずれも第4章で説明した手順でシミュレーションできるので，手元で試してみてください。ソフトウェアと同じようにプログラミングしてハードウェアを設

計できるという雰囲気を感じましょう。

## 1. 基本的なハードウェア ——4ビット・カウンタのプログラム

カウンタは，基本的なハードウェア・モジュールの一つです。入力された信号の変化に応じて内部の変数をインクリメント(1ずつ値を増加)します。使用法は，クロックの立ち上がりに応じて内部の変数をカウントし，そのカウント回数により，処理のタイミングを遅らせたり，定期的に

リスト1 指定ビット数のカウンタ (VHDL版)

```

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;
use IEEE.std_logic_arith.all;

entity counter is
    generic (
        width : integer := 4
    );
    port (
        pClk : in std_logic;
        Q : out std_logic_vector(width-1 downto 0);
        pReset : in std_logic;
    end counter;

    architecture rtl of counter is
        signal counter : std_logic_vector(width-1 downto 0);
    begin
        Q <= counter;

        process (pClk, pReset)
        begin
            if pReset = '1' then
                counter <= (others => '0');
            elsif pClk'event and pClk = '1' then
                counter <= counter + 1;
            end if;
        end process;
    end rtl;
    
```

リスト2 指定ビット数のカウンタ (Verilog HDL版)

```

module counter(pClk, pReset, Q);
    parameter width = 4;

    input pClk;
    input pReset;

    output [width-1:0] Q;

    reg [width-1:0] counter;

    assign Q = counter;

    always @(posedge pClk or posedge pReset) begin
        if(pReset) begin /* pReset=1 */
            counter <= 0;
        end else begin
            counter <= counter + 1;
        end
    end
endmodule
    
```

注1: signedを指定しない場合は，基本的にunsignedになる  
 注2: always文はかっこ内に変化を監視する信号を記述する  
 posedgeは立ち上がりの瞬間に相当する