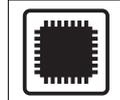


# FPGAで学ぶVerilog HDL

第3回

180秒のタイマを作る



デバイスの記事



ビギナーズ



関連データ

井倉将実



前回(2005年10月号, pp.130-134)は、1秒ごとにLEDが点灯/消灯を繰り返す(点滅する)回路を設計しました。この回路で「1秒ごとに動作する回路」を作ることができたわけです。その応用として、今回は台所で使うキッチン・タイマを作ってみることにしましょう。なお、本記事で紹介する設計データは本誌Webサイト(<http://www.cqpub.co.jp/dwm/>)からダウンロードすることができます。(筆者)

前回までで1秒ごとに動作する回路を設計できました。そこで最終的に時計を設計することを考えていきましょう。1秒ごとの動作からいきなり完全な時計を作るのはまだ壁が高そうです。そこで今回は、3分間クッキング用の180秒キッチン・タイマを作ります。

## 1. 180秒をカウントする

このキッチン・タイマは、180秒という値があらかじめセットされているものです。また、残り時間を3けたの7セグメントLEDに表示します。そしてスタート・ボタンを押すと1秒ごとにLEDの数字が減少し、180秒後には内部カウンタは0秒になります。このとき人間の目では、始まりの180から、179, 178, 177, ..., 000と、1秒ごとに減少するように見えます。0秒になると7セグメントLEDに「-」を点滅し、180秒が経過したことを示します。タイマ動作中にリセット・ボタンを押すと、タイマの値が180秒に戻り、次のスタート・ボタンが押されるまで待ち続けます。

図1に、設計するキッチン・タイマの動作をまとめます。リスト1は、180秒キッチン・タイマのVerilog HDL記述です。宣言部(モジュール宣言, ポート宣言, レジスタ宣言)が大きくなっていますが、行数が増えているだけで新

しいことはありません。

記述は大きく五つの処理に分かれています。

- タイマ駆動開始フラグ生成
- 1秒タイミング生成
- 180秒タイマ回路
- タイムアウト検出回路
- 7セグメントLED表示部

### ● タイマ駆動開始フラグ生成——スイッチの処理

スイッチの押下状態によって、タイマのスタートと、タイマをリセットする機能を実現します。

ここでは、今回作る回路のいちばん重要な部分であるタイマ回路に対して適用される、二つのスイッチ状態に応じた「タイマ動作有効フラグ」を作っています(図2)。

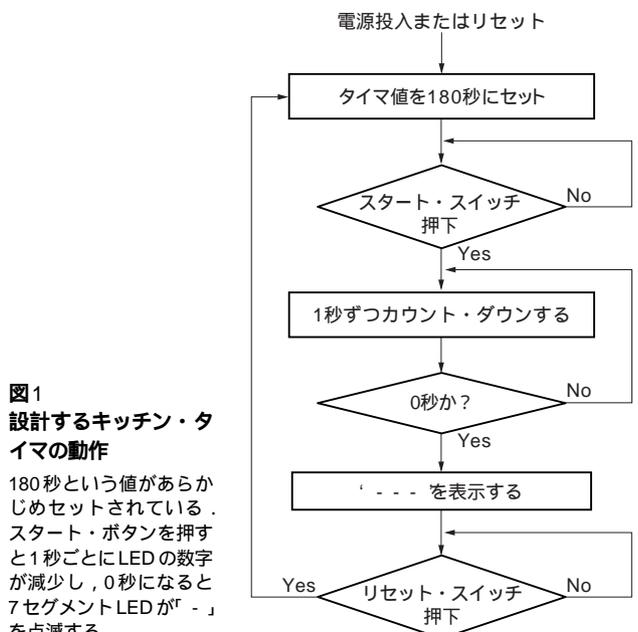


図1 設計するキッチン・タイマの動作

180秒という値があらかじめセットされている。スタート・ボタンを押すと1秒ごとにLEDの数字が減少し、0秒になると7セグメントLEDが「-」を点滅する。

タイマ動作有効フラグは、スタート・ボタンの押下によって有効(“H”)になり、リセット・ボタンの押下で無効(“L”)になります。タイマ回路には、このフラグが有効なときだけ動作させるようなしくみを持たせます。

ところで「スタート・ボタンを押したとき“H”、リセット・ボタンを押したとき“L”」という動作を単純に記述すると、リスト2(a)のようになります。しかし、この記述には落とし穴があります。スタート・ボタンとリセット・ボタンを同時に押したときに問題が起こるのです。

このような場合、どちらか一方のスイッチ入力を優先するように記述します。スイッチが同時に押されたときには、優先したスイッチの動作を実行します。

リセット・スイッチを優先した記述をリスト2(b)に示します。二つのスイッチが同時に押されたときには、タイマ動作有効フラグは無効(“L”)になります。



図2 タイマ動作有効フラグ

スタート・スイッチを押す(“L”になる)と、タイマ動作有効フラグが“H”になる。リセット・スイッチを押すと、タイマ動作有効フラグが“L”になる。

## ● 1秒タイミング生成回路

1秒タイミング生成回路は前回と同じ回路です。1秒ごとに1クロック分の幅だけ有効なフラグ信号(Pulse\_1sec)が出力されます。

タイマ動作有効フラグ(TIM\_START)が有効(“H”)のときのみ動作するような記述が追加されています。

### リスト2 スタート・ボタンを押したとき“H”、リセット・ボタンを押したとき“L”にするVerilog HDL記述

```
always @( posedge SYSCLK )
begin
    if ( SW_START == 1'b0 ) begin
        TIM_START <= 1'b1 ;
    end

    if ( SW_RESET == 1'b0 ) begin
        TIM_START <= 1'b0 ;
    end
end
```

(a) 正しく動作しない記述

```
always @( posedge SYSCLK )
begin
    if ( SW_RESET == 1'b0 ) begin
        // リセット・スイッチが押されたら
        TIM_START <= 1'b0 ;
        // -> タイマ駆動開始フラグをクリア
    end else begin // リセット・スイッチが押されておらず
        if ( SW_START == 1'b0 ) begin
            // スタート・スイッチが押されたら
            TIM_START <= 1'b1 ;
            // -> タイマ駆動開始フラグをセット
        end
    end
end
```

(b) リセット・スイッチを優先させる記述

### リスト1 180秒キッチン・タイマのVerilog HDL記述

```
module TIM180(
// 入力ピン
    SYSCLK,
    SW_START,
    SW_RESET,

// 出力ピン
    SEG_1,
    SEG_2,
    SEG_3
);

// 入出力ピンの定義
    input          SYSCLK ;
    input          SW_START ;
    input          SW_RESET ;

    output [7:0]   SEG_1 ;
    output [7:0]   SEG_2 ;
    output [7:0]   SEG_3 ;

    wire [7:0]     SEG_1 ;
    wire [7:0]     SEG_2 ;
    wire [7:0]     SEG_3 ;

// 内部回路用のレジスタ宣言
    parameter [31:0] param_1Second = 32'h01F78A40 ;
    // 33MHz時の1秒タイミング

    reg [31:0] Sec1_counter ;

    reg [ 3:0] TIM_3 ; // 100秒けたタイマ
    reg [ 3:0] TIM_2 ; // 10秒けたタイマ
    reg [ 3:0] TIM_1 ; // 1秒けたタイマ

    reg [ 7:0] node_SEG3 ; // 7Seg用内部ノード
    reg [ 7:0] node_SEG2 ; // 7Seg用内部ノード
    reg [ 7:0] node_SEG1 ; // 7Seg用内部ノード

    reg Pulse_1Sec ;
    reg TIM_START ;
    reg TIMEOUT ;

    /*****
    // タイマ駆動開始フラグ生成
    *****/
    always @( posedge SYSCLK )
begin
    if ( SW_RESET == 1'b0 ) begin
        // リセット・スイッチが押されたら
```