

1桁のプログラムを参考に、2桁に拡張します。赤字で記入した部分を追加するだけです。

```
module bin2dec (CLOCK_50, KEY, SW, HEX0, HEX1);
```

```
    input CLOCK_50; // ボードクロック 50MHz
```

```
    input [3:0] KEY; // rstn (リセット) 用プッシュスイッチ
```

```
    input [17:0] SW; // 2進数入力用スライドスイッチ
```

```
    output [6:0] HEX0, HEX1; // 出力用 7 セグ LED
```

ポート宣言に使用したものは、すべて module bin2dec() の括弧の中に記述しなければなりません。

```
seg7_decimal_1digit u0(.clk(CLOCK_50), .rstn(KEY[0]), .in(SW[6:0]), .out0(HEX0), .out1(HEX1));
```

```
endmodule
```

今回は 7 ビット入力ですが、2桁 (99) までしか表示できないため、1100011 (99 の 2 進数表記) 以降の入力は正しく表示されません。

7 セグメントディスプレイに表示させるためのプログラムです。d0 は一番右端の **HEX0** を光らせ、d1 はその隣の **HEX1** を光らせるためのプログラムとなります。

```
module seg7_decimal_1digit (clk, rstn, in, out0, out1);
```

```
    input clk, rstn;
```

```
    input [6:0] in; // 10 進数に変換したい 2 進数
```

```
    output [6:0] out0, out1; // 各 7 セグ LED が表示すべき 10 進数
```

```
    reg [6:0] tmp_reg; // 変換用計算用レジスタ
```

```
    reg [3:0] digit0, digit1; // 各 7 セグ LED が表示すべき 10 進数
```

```
    seg7_decorder d0(.in(digit0), .out(out0));
```

```
    seg7_decorder d1(.in(digit1), .out(out1));
```

```
    always @(posedge clk) begin
```

```
        if (!rstn) begin
```

```
            tmp_reg <= in;
```

```
            digit0 <= 4'd0;
```

```
            digit1 <= 4'd0;
```

```
        end else begin
```

rstn=0 (リセットボタンが押されたら)、tmp_reg に in の値、digit0 に 0、digit1 に 0 が代入されます。このプログラムでは、リセットボタンを押さないとセグメントに表示がされないため、注意が必要です。

```

    if (tmp_reg >= 7'd10) begin // 10 の位を計算
        tmp_reg <= tmp_reg - 7'd10;
        digit1 <= digit1 + 1'd1;
    end else begin //残った 1 の位を代入
        digit0 <= tmp_reg;
    end
end
end
end
endmodule

```

tmp_reg (一番最初はスイッチの入力) が 10 より大きいとき以下を計算します。

たとえば「35 (100011)」のとき、
 tmp_reg=35 で 10 よりも大きいから
 tmp_reg = 35-10 = 25
 digit1 = digit1+1=0+1 なので、1 を入れる

tmp_reg=25 なので、10 より大きい
 tmp_reg = 25-10 = 15
 digit1 = 1+1=2

tmp_reg=15 なので、10 より大きい
 tmp_reg = 15-10 = 5
 digit1 = 2+1=3

tmp_reg=5 なので、10 より小さい
 else 以降を計算

```

module seg7_decorder (in, out);

```

```

    input [3:0] in;

```

```

    output [6:0] out;

```

このプログラムは変更しないでください。

```

    assign out =      (in == 4'd0)?   7'b1000000 :
                      (in == 4'd1)?   7'b1111001 :
                      (in == 4'd2)?   7'b0100100 :
                      (in == 4'd3)?   7'b0110000 :
                      (in == 4'd4)?   7'b0011001 :
                      (in == 4'd5)?   7'b0010010 :
                      (in == 4'd6)?   7'b0000010 :
                      (in == 4'd7)?   7'b1111000 :
                      (in == 4'd8)?   7'b0000000 :
                      (in == 4'd9)?   7'b0010000 : 7'b1111111;

```

```

endmodule

```

7-segment Displays

