

実習 4-2 2進数を 10 進数に表示するプログラム (4 桁)

前回の 2 桁を 4 桁に拡張します。青字で記入した部分を追加するだけで対応できます。やっていることは 2 桁のときとほとんど変わりません。なお、module seg7_decoder は変更がないため、この紙面では省略します。

```
module bin2dec (CLOCK_50, KEY, SW, HEX0, HEX1, HEX2, HEX3);

    input CLOCK_50; // ボードクロック 50MHz

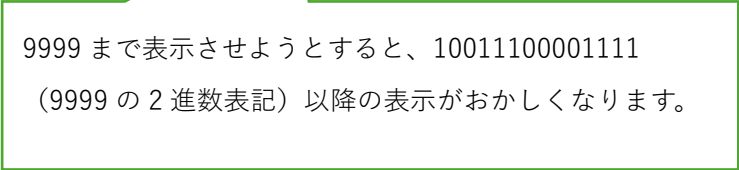
    input [3:0] KEY; // rstn (リセット) 用プッシュスイッチ

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

    output [6:0] HEX0, HEX1, HEX2, HEX3; // 出力用 7 セグ LED

    seg7_decimal_1digit
    u0(.clk(CLOCK_50), .rstn(KEY[0]), .in(SW[13:0]), .out0(HEX0), .out1(HEX1), .out2(HEX2), .out3(HEX3));

endmodule
```



9999 まで表示させようとする、10011100001111 (9999 の 2 進数表記) 以降の表示がおかしくなります。

```
module seg7_decimal_1digit (clk, rstn, in, out0, out1, out2, out3);

    input clk, rstn;

    input [13:0] in; // 10 進数に変換したい 2 進数

    output [6:0] out0, out1, out2, out3; // 各 7 セグ LED が表示すべき 10 進数

    reg [13:0] tmp_reg; // 変換用計算用レジスタ

    reg [3:0] digit0, digit1, digit2, digit3; // 各 7 セグ LED が表示すべき 10 進数

    seg7_decoder d0(.in(digit0), .out(out0));

    seg7_decoder d1(.in(digit1), .out(out1));

    seg7_decoder d2(.in(digit2), .out(out2));

    seg7_decoder d3(.in(digit3), .out(out3));
```

```

always @(posedge clk) begin

    if (!rstn) begin

        tmp_reg <= in;

        digit0 <= 4'd0;

        digit1 <= 4'd0;

        digit2 <= 4'd0;

        digit3 <= 4'd0;

    end

    else

        if (tmp_reg >= 14'd1000) begin // 千の位を計算

            tmp_reg <= tmp_reg - 14'd1000;

            digit3 <= digit3 + 1'd1;

        end

        else

            if (tmp_reg >= 10'd100) begin // 百の位を計算

                tmp_reg <= tmp_reg - 10'd100;

                digit2 <= digit2 + 1'd1;

            end

            else

                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
1000 より大きいとき以下を計算

例えば「2539 (100111101011)」のとき、

tmp_reg=2539 で 1000 よりも大きいから

tmp_reg = 2539 - 1000 = 1539

digit3 = digit3 + 1 = 0 + 1 なので、1 を入れる

tmp_reg = 1539 なので、1000 より大きい

tmp_reg = 1539 - 1000 = 539

digit3 = 1 + 1 = 2

tmp_reg = 539 なので、1000 より小さい

else 以降を計算

tmp_reg = 539 なので、100 より大きい

tmp_reg = 539 - 100 = 439

digit2 = 0 + 1 = 1

・・・とつづけて同じことをしていくと

digit3 = 2

digit2 = 5

digit1 = 3

digit0 = 0

が入るようになります。