

課題23-4

コード

TopModule.v

```
module TopModule(
    //////////////// CLOCK ///////////
    input          CLK1,
    input          CLK2,
    //////////////// SEG7 ///////////
    output [7:0]   HEX0,
    output [7:0]   HEX1,
    output [7:0]   HEX2,
    output [7:0]   HEX3,
    output [7:0]   HEX4,
    output [7:0]   HEX5,
    //////////////// Push Button ///////////
    input [1:0]    BTN,
    //////////////// LED ///////////
    output [9:0]   LED,
    //////////////// SW ///////////
    input [9:0]   SW,
    //////////////// Matrix Key ///////////
    input [3:0]    KEY_ROW,
    output [3:0]   KEY_COL
);

wire clk;                      // 100Hzに分周されたクロック
wire [3:0] wq;                // 押下キー(4bit)
wire [3:0] num;                // 表示用に変換されたキー値
wire [15:0] key;              // マトリクスキーの16bit出力
reg [3:0] rdata0, rdata1, rdata2, rdata3, rdata4, rdata5; // 入力履歴保存

レジスタ
wire pushed;                  // 打鍵検出信号
reg pushed_d;                 // pushedの1クロック遅延

m_prescale(CLK1, clk);        // 50MHz → 100Hz 分周

m_matrix_key(clk, SW[0], KEY_ROW, KEY_COL, key, tc); // マトリクスキー走査

m_dec16to4(key, wq, pushed); // 押下キーを4bitに変換

m_convert_num(wq, num);      // キー配列に応じた数値変換

// pushedの立ち上がり検出用遅延
always @(posedge clk) begin
    pushed_d <= pushed;
end
```

```

// 新しいキーが押されたときのみ履歴をシフト
always @(posedge clk) begin
    if (!pushed_d && pushed) begin
        rdata5 <= rdata4;
        rdata4 <= rdata3;
        rdata3 <= rdata2;
        rdata2 <= rdata1;
        rdata1 <= rdata0;
        rdata0 <= num;      // 最新の入力
    end
end

wire [7:0] dec0, dec1, dec2, dec3, dec4, dec5;

// 各履歴データを7セグ表示用にデコード
m_7segment u0 (rdata0, dec0);
m_7segment u1 (rdata1, dec1);
m_7segment u2 (rdata2, dec2);
m_7segment u3 (rdata3, dec3);
m_7segment u4 (rdata4, dec4);
m_7segment u5 (rdata5, dec5);

assign LED = {6'd0,wq};      // 押下キーをLEDで表示
assign HEX0 = dec0;
assign HEX1 = dec1;
assign HEX2 = dec2;
assign HEX3 = dec3;
assign HEX4 = dec4;
assign HEX5 = dec5;

endmodule

```

動作確認

- ボタンを押すと、7セグメントLEDに押したキーの番号が表示されることを確認した。
- 直近6回分の入力が左から新しい順に表示されることを確認した。
- 左にシフトしていくことを確認した。

解説

マトリクスキーで入力された数値を順に記録し、直近6回分の入力を7セグメントLEDに表示する構成になっている。

まず `m_prescale` により基板のクロックを 100Hz に分周し、マトリクスキーの走査や入力処理を人間にとつて安定して扱える速度にしている。`m_matrix_key` は列を順番に Low にしながら行の入力を読み取り、16個のキー状態を 16bit の `key` 信号として出力する。

`m_dec16to4` では `key` のうち、1つだけ押されている場合にその位置を 4bit の `wq` として取り出し、同時に `pushed` 信号で打鍵を検出する。`m_convert_num` はマトリクスキーの物理配置に合わせて、`wq` を実際に表

示したい数値へ変換する役割を持つ。

入力履歴の更新では `pushed` の立ち上がりのみを検出するため、`pushed_d` に1クロック遅延させた信号を用いている。`pushed` が 0 から 1 になった瞬間だけ、過去のデータを `rdata1`～`rdata5` にシフトし、最新の入力を `rdata0` に保存する。これによりキーを押し続けても同じ値が連続して記録されることを防いでいる。

最後に `rdata0` から `rdata5` に保存された6個のデータをそれぞれ7セグメントデコーダに渡し、HEX0 から HEX5 に表示する。これにより、マトリクスキーの入力履歴が左から新しい順に並んで表示される。