Распознавание звукового тона методом zero-crossing.

Интерфейс SPI.

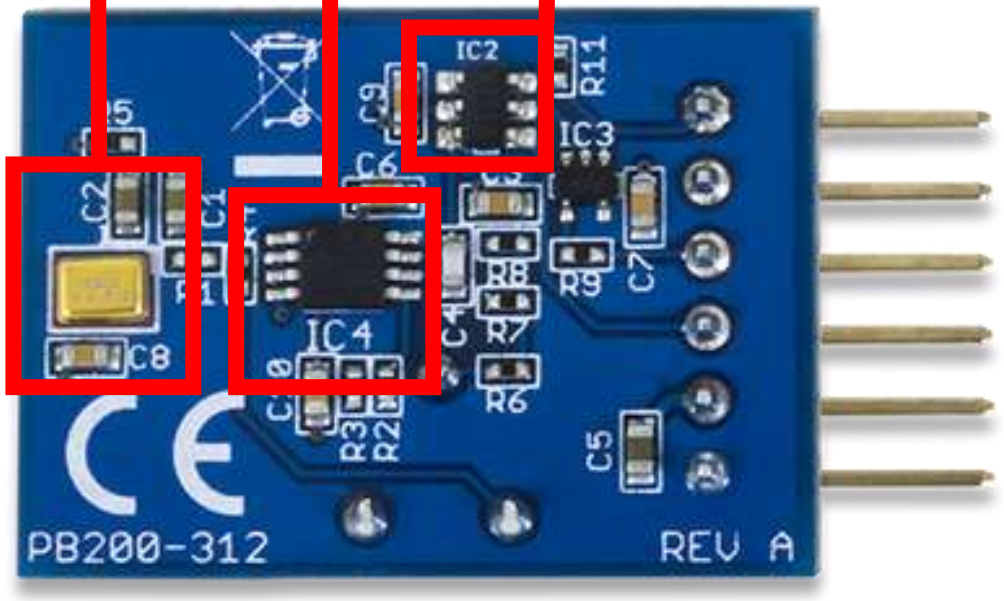
Digilent PMOD MIC3

Регулировка чувствительности

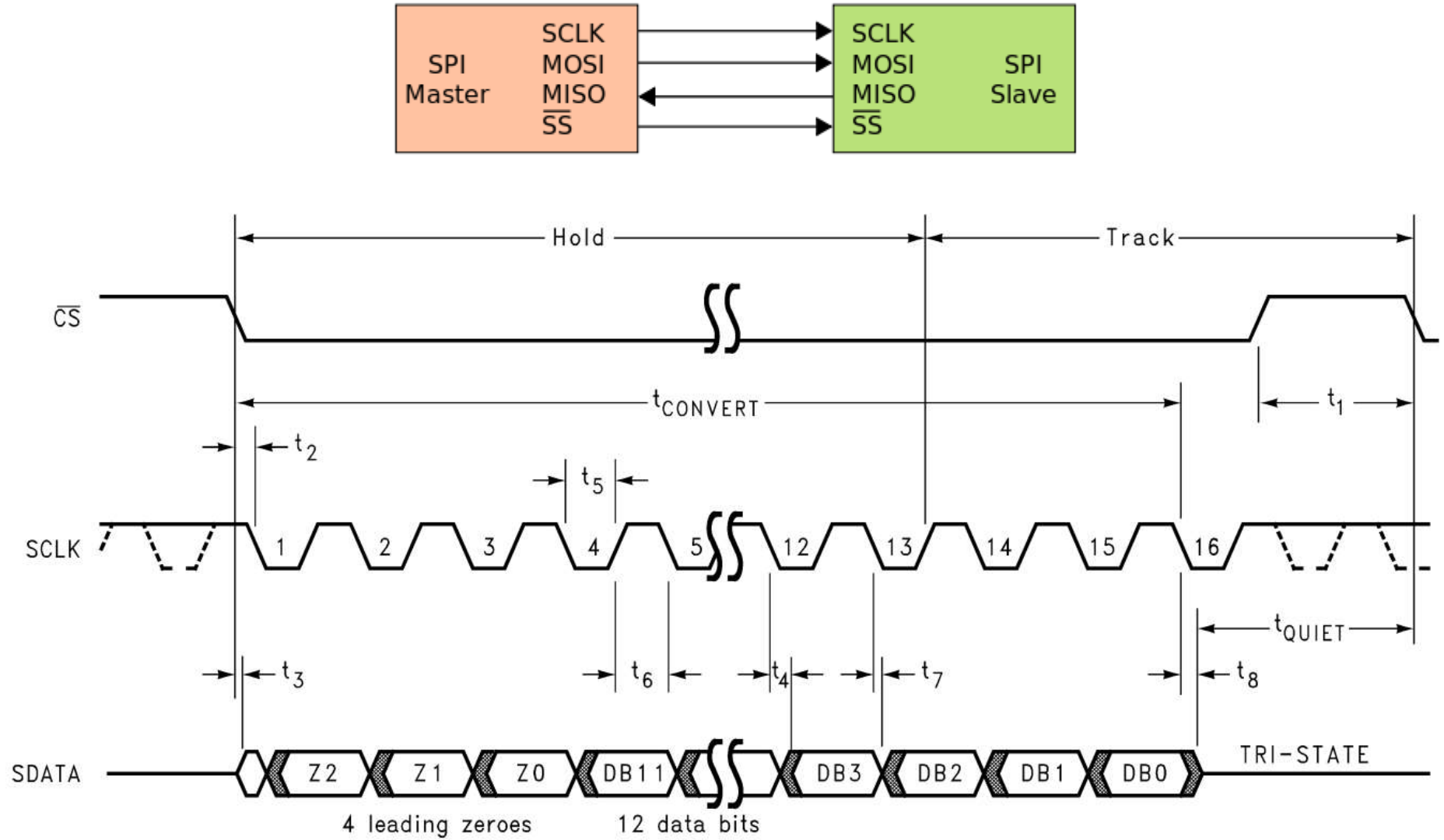
Микрофон

Усилитель

12-битный послед. АЦП



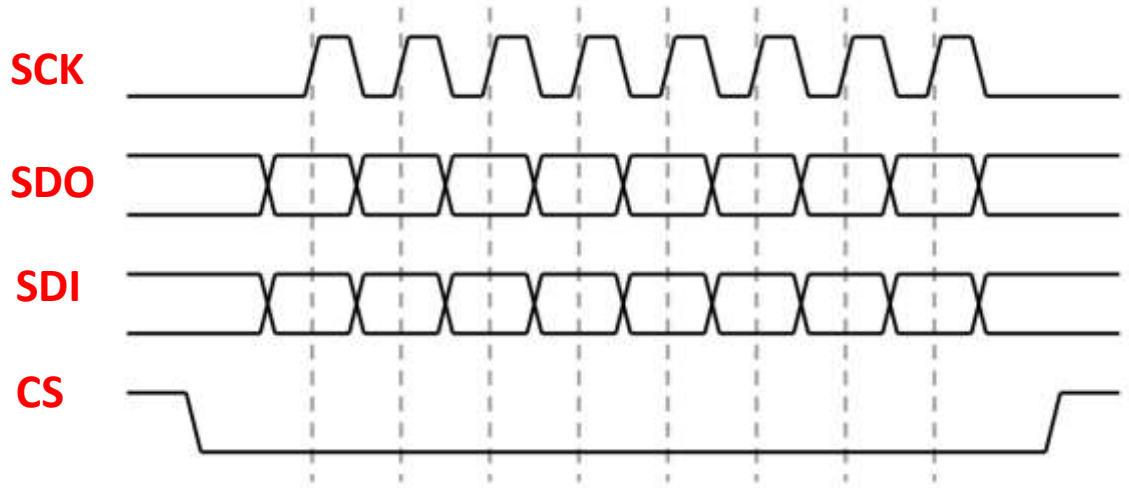
Интерфейс SPI



Реализация интерфейса SPI на Verilog

```
3.  module pmod_mic3_spi_receiver
4.  (
5.      input           clock,  // Main Clock 50 MHZ
6.      input           reset,
7.      output          cs,     // Chip Select
8.      output          sck,    // Serial Clock
9.      input           sdo,    // Serial data output
10.     output reg [15:0] value
11. );
```

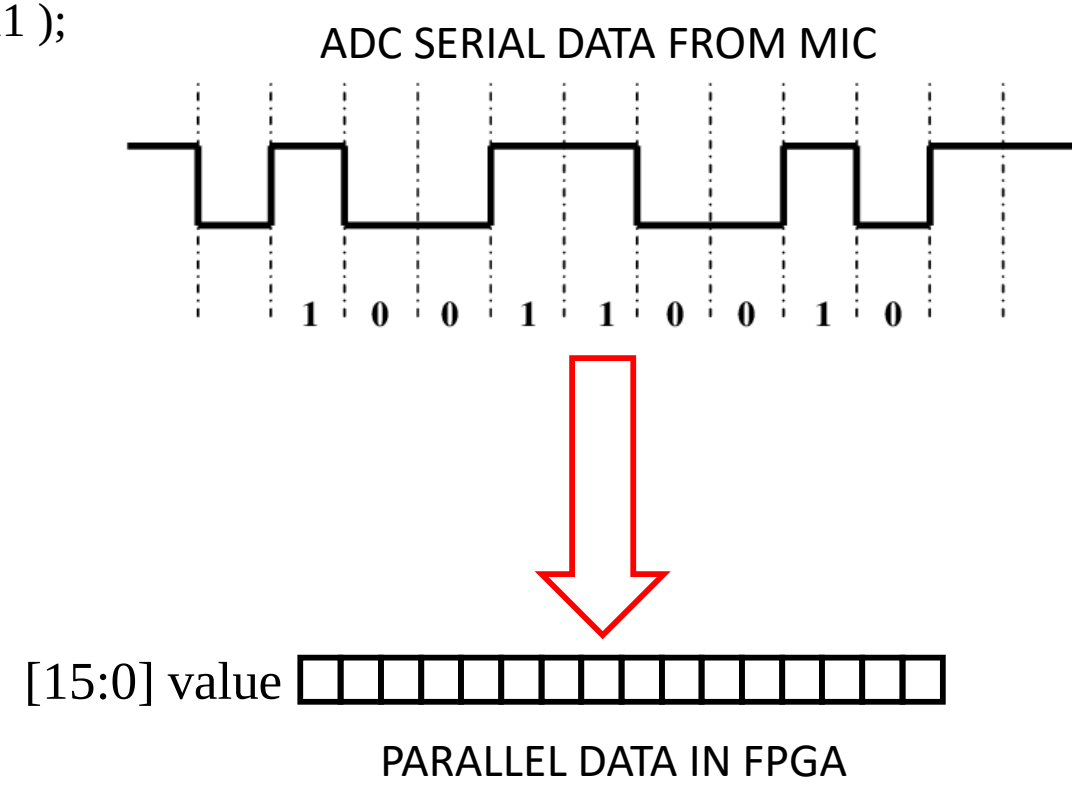
```
16.  always @ (posedge clock or posedge reset)
17.    begin
18.      if (reset)
19.        cnt <= 7'b100;
20.      else
21.        cnt <= cnt + 7'b1;
22.    end
23.    assign sck = ~ cnt [1]; // 12.5 MHz
24.    assign cs  =   cnt [6]; // 390.625 KHz
```



day_2/common/pmod_mic3_spi_receiver.v

Реализация интерфейса SPI на Verilog

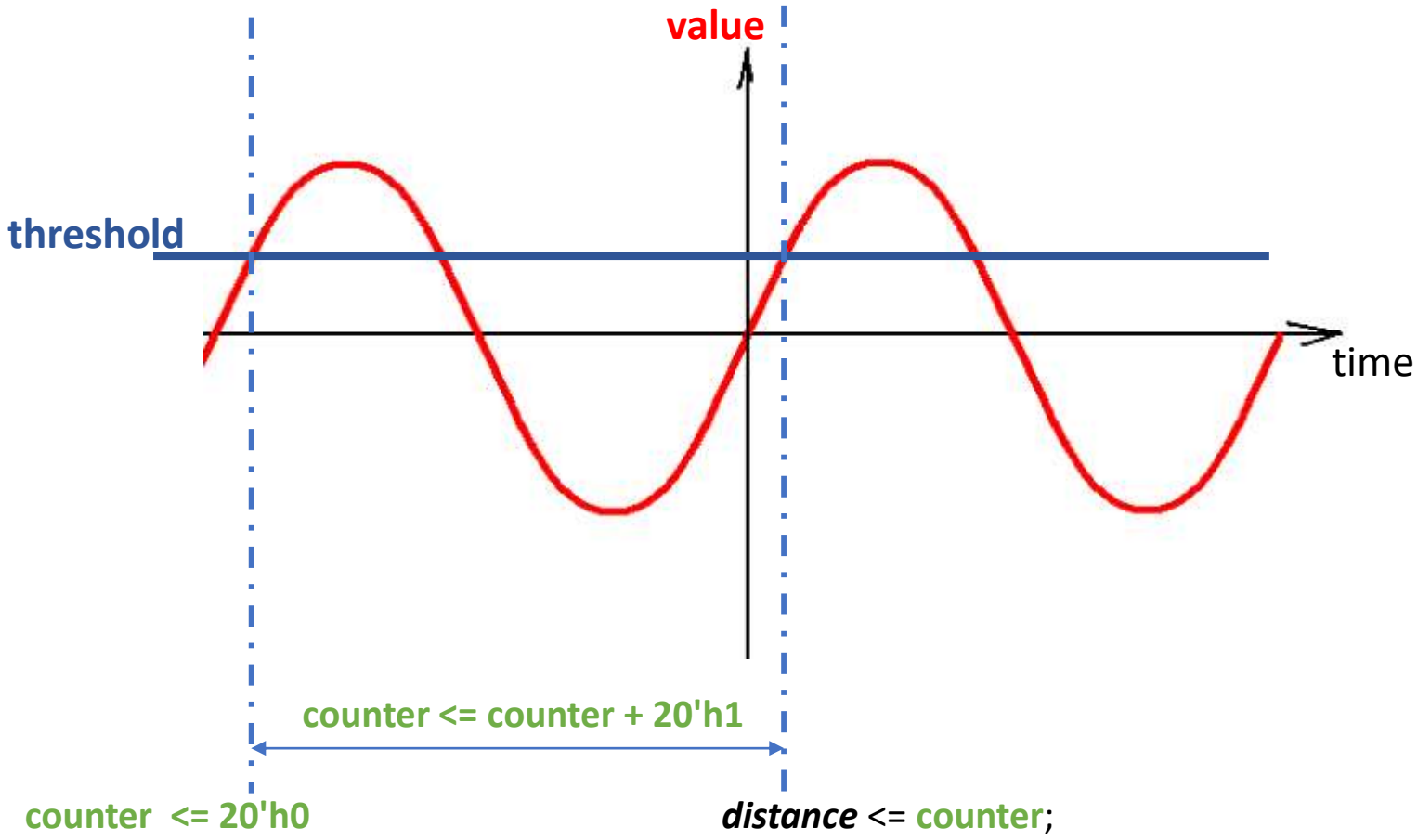
```
27.   wire sample_bit = ( cs == 1'b0 && cnt [1:0] == 2'b11 );
28.   wire value_done = ( cnt [6:0] == 7'b0 );
29.
30.   always @ (posedge clock or posedge reset)
31.   begin
32.      if (reset)
33.      begin
34.         shift <= 16'h0000;
35.         value <= 16'h0000;
36.      end
37.      else if (sample_bit)
38.      begin
39.         shift <= (shift << 1) | sdo;
40.      end
41.      else if (value_done)
42.      begin
43.         value <= shift; // 16-bit parallel data from mic
44.      end
45.   end
```



day_2/common/pmod_mic3_spi_receiver.v

Распознавание звукового тона с помощью “zero-crossing”

```
56.  localparam [15:0] threshold = 16'h1100;
57.    always @ (posedge clk or posedge reset)
58.       if (reset)
59.       begin
60.          prev_value <= 16'h0;
61.          counter    <= 20'h0;
62.          distance   <= 20'h0;
63.       end
64.       else
65.       begin
66.          prev_value <= value;
67.          if ( value     >= threshold
68.             & prev_value < threshold)
69.          begin
70.            distance <= counter;
71.            counter  <= 20'h0;
72.          end
73.          else if (counter != ~ 20'h0)
74.          begin
75.            counter <= counter + 20'h1;
76.          end
77.       end
```



Отсчитываем число тактов в **counter** после пересечения **treshhold**, до следующего пересечения threshold и записываем это число в **distance.**

day_2//lab_5_mic/top.v*

Интерпретация полученных данных

*Частоты нот в Гц*100*

```
localparam
        freq_100_C  = 26163,
        freq_100_Cs = 27718,
        freq_100_D  = 29366,
        freq_100_Ds = 31113,
        freq_100_E  = 32963,
        freq_100_F  = 34923,
        freq_100_Fs = 36999,
        freq_100_G  = 39200,
        freq_100_Gs = 41530,
        freq_100_A  = 44000,
        freq_100_As = 46616,
        freq_100_B  = 49388;
```

```
function [19:0] high_distance (input [18:0] freq_100);          // Проверка по верхнему порогу частоты
    high_distance = clk_mhz * 1000 * 1000 / freq_100 * 102;  //Перевод частоты ноты в число тактов distance
endfunction
  //----------------------------------------------------------------------

  function [19:0] low_distance (input [18:0] freq_100);         // Проверка по нижнему порогу частоты
    low_distance = clk_mhz * 1000 * 1000 / freq_100 * 98;    //Перевод частоты ноты в число тактов distance
endfunction

  //----------------------------------------------------------------------

  function [19:0] check_freq_single_range (input [18:0] freq_100); // Проверка соответствия distance
                                                                   // заданной частоте ноты
check_freq_single_range =   distance > low_distance  (freq_100)
                          & distance < high_distance (freq_100);
  endfunction

  //----------------------------------------------------------------------

  function [19:0] check_freq (input [18:0] freq_100);              // Проверка по следующим двум октавам

    check_freq =  check_freq_single_range (freq_100 * 4)
                | check_freq_single_range (freq_100 * 2)
                | check_freq_single_range (freq_100);

  endfunction
```

day_2//lab_5_mic/top.v (93-153)*

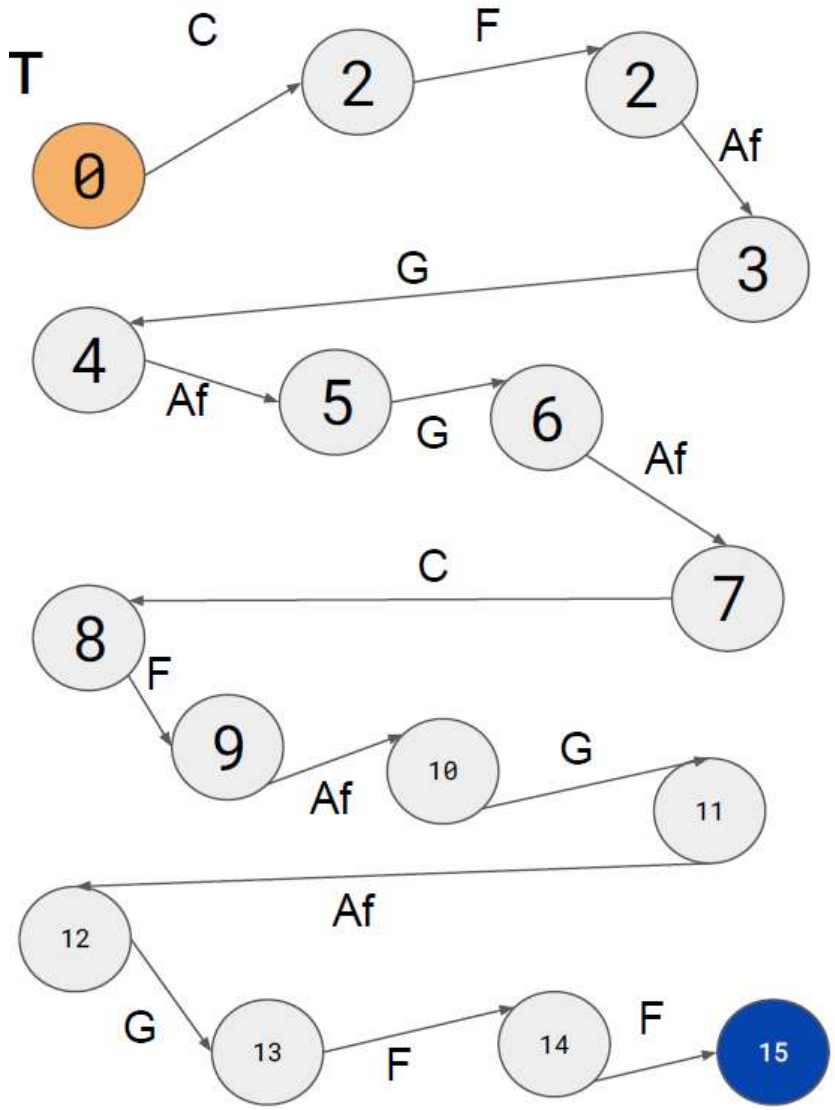
Вывод ноты на семисегментный индикатор

```
438     always @ (posedge clk or posedge reset)
439         if (reset)
440         begin
441             abcdefgh <= 8'b11111111;
442         end
443         else if (digit_enable)
444         begin
445             if (i_digit == 3'd3)
446                 case (t_note)
447                 C  : abcdefgh <= 8'b01100011;  // C   // abcdefgh
448                 Cs : abcdefgh <= 8'b01100010;  // C#
449                 D  : abcdefgh <= 8'b10000101;  // D   //   --a--
450                 Ds : abcdefgh <= 8'b10000100;  // D#  //  |     |
451                 E  : abcdefgh <= 8'b01100001;  // E   //  f     b
452                 F  : abcdefgh <= 8'b01110001;  // F   //  |     |
453                 Fs : abcdefgh <= 8'b01110000;  // F#  //   --g--
454                 G  : abcdefgh <= 8'b01000011;  // G   //  |     |
455                 Gs : abcdefgh <= 8'b01000010;  // G#  //  e     c
456                 A  : abcdefgh <= 8'b00010001;  // A   //  |     |
457                 As : abcdefgh <= 8'b00010000;  // A#  //   --d--  h
458                 B  : abcdefgh <= 8'b11000001;  // B
459                 default : abcdefgh <= 8'b11111111;
460                 endcase
```

day_2//lab_5_mic/top.v*

А как распознавать не отдельные ноты, а мелодии?

```
always @ (posedge clk or posedge reset)
    if (reset)
      states [2] <= 0;
    else
      case (states [2])
       0: if ( t_note == G  ) states [2] <=  1;
       1: if ( t_note == F  ) states [2] <=  2;
       2: if ( t_note == A  ) states [2] <=  3;
       3: if ( t_note == B  ) states [2] <=  4;
       4: if ( t_note == Cs ) states [2] <=  5;
       5: if ( t_note == D  ) states [2] <=  6;
       6: if ( t_note == E  ) states [2] <=  7;
       7: if ( t_note == F  ) states [2] <=  8;
       8: if ( t_note == E  ) states [2] <=  9;
       9: if ( t_note == D  ) states [2] <= 10;
      10: if ( t_note == C  ) states [2] <= 11;
      11: if ( t_note == Bf ) states [2] <= 12;
      12: if ( t_note == A  ) states [2] <= 13;
      13: if ( t_note == G  ) states [2] <= 14;
      14: if ( t_note == Bf ) states [2] <= recognized;
      endcase
```



day_2//lab_5_mic/top.v (375-395)*

Подключение микрофона к плате rzrd/omdazz