

```
////////////////////////////////////
////////////////////////////////////
//
// Company:
// Engineer:      Ralnikov Vadim Dmitryevich
//
// Create Date:   15:00 23/02/2019
// Design Name:
// Module Name:   shiftreg8_1
// Project Name:
// Target Devices:
// Tool versions:
// Description:    shift reg 8 parralel in - 1 out
// sync, with async reset.
//
////////////////////////////////////
////////////////////////////////////

`timescale 1ns / 100ps
`default_nettype none

/*
shift_reg8_1
  shift_reg8_1_inst
  (
    .CLK                      (),
    // in, u[ 1], Clock freq 100MHz
    .SLOAD                    (),
    // in, u[ 1], register sync loading
    .EN                        (),
    // in, u[ 1], enable work 1 -enable
    .DIR                       (),
    // in, u[ 1], direction setter, 1 - left, 0
    // - right
    .DATA                      (),
    // in, u[ 8], input data, parallel
    .RST                       (),
    // in, u[ 1], reset input, 0 - reset async

    .Q                         (),
```

```
        // out, u[ 1], normal polarity
    )
*/

module shiftreg8_1
(
    input  wire          CLK,
    // in, u[ 1], Clock freq 100MHz
    input  wire          SLOAD,
    // in, u[ 1], register sync loading
    input  wire          EN,
    // in, u[ 1], enable work 1 -enable
    input  wire          DIR,
    // in, u[ 1], direction setter, 1 - left, 0
    // - right
    input  wire [ 7 : 0 ] DATA,
    // in, u[ 8], input data, parallel
    input  wire          RST,
    // in, u[ 1], reset input, 0 - reset async

    output wire          Q
    // out, u[ 1], normal polarity
);

reg [ 7 : 0 ] r_sh = 8'b0;
// reg, u[ 8], for shifting

assign Q = DIR ? r_sh[7] : r_sh[0];
// output selector left or right

parameter LEFT  = 1'b1;
parameter RIGHT = 1'b0;

always @(posedge CLK or negedge RST)
```

```
// cycle start
  if(~RST)
    // async reset 0 - reset
    r_sh <=0;
  else begin
    if(EN)
      // enable inicial 1 - enable
      if(SLOAD)
        r_sh <= DATA;
      else
        case(DIR)
          // direction select 1 left 0
          right
          LEFT:      r_sh <= {r_sh[ 6 :
0 ], 1'b0};
          RIGHT:     r_sh <= {1'b0, r_sh[
7 : 1 ]};

          default:   r_sh <= {r_sh[ 6 :
0 ], 1'b0};
        endcase
      end
    end
  end
endmodule
```