`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company: Ratner Surf Designs

// Engineer: James Ratner

//

// Create Date: 07/04/2018 02:46:31 PM

// Design Name:

// Module Name: usr\_nb

// Project Name:

// Target Devices:

// Tool Versions:

// Description: Generic n-bit universial shift register

// with a asynchronous positive logic reset.

//

//

// SEL | Operation

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

// 00 | Hold

// 01 | Load (input data\_in)

// 10 | Shift left (input dbit on right)

// 11 | Shift right (input dbit on left)

//

// usr\_nb #(.n(16)) MY\_USR (

// .data\_in (my\_data\_in),

// .dbit (my\_dbit),

// .sel (my\_sel),

// .clk (my\_clk),

// .clr (my\_clr),

// .data\_out (my\_data\_out)

// );

//

// Dependencies:

//

// Revision:

// Revision 1.00 - File Created (07-06-2018)

//

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module usr\_nb(data\_in, dbit, sel, clk, clr, data\_out);

input [n-1:0] data\_in;

input dbit, clk, clr;

input [1:0] sel;

output reg [n-1:0] data\_out;

parameter n = 8;

always @(posedge clr, posedge clk)

begin

if (clr == 1) // asynch reset

data\_out <= 0;

else

case (sel)

0: data\_out <= data\_out; // hold value

1: data\_out <= data\_in; // load

2: data\_out <= {data\_out[n-2:0],dbit}; // shift left

3: data\_out <= {dbit,data\_out[n-1:1]}; // shift right

default data\_out <= 0;

endcase

end

endmodule