```
/*
   Ben Davis
   2/13/24
   EE 371
   Lab 4, Task 1

   This module takes in an 8 bit input and outputs the number of
   ones in the input data once it is done. It has a start enable
   input, a shift right input, and an output zero when it is finished.

*/

module lab4task1_ASM (
               input logic clk, //clock
               input logic [7:0] in, //data input
               input logic start, //start enable
               input logic shr, //shift enable
               output logic zero, //the data has all 0's
               output logic [3:0] result //number of 1's in the data
               );


   logic [7:0] A; //intermediate data for the input

   always_ff @(posedge clk) begin
      /* continuously reset the result to zero
      and A to the input if the start enable is off*/
      if(~start) begin
         result <= 4'b0;
         A <= in;
      /*once started, checks if the lsb in A is equal
      to 1 and adds one to the counter if true. Then,
      regardless, it shifts right*/
      end else if(start) begin
         if((A[0])==(1'b1)) begin
            result <= result + 1'b1;
         end
         zero <= ((A[7:0])==(8'b00000000));
         A <= A >> 1;
      end
   end
endmodule
//testbench
module lab4task1_ASM_tb ();

   //recall variables
   logic clk, start, shr, zero;
   logic [7:0] in;
   logic [3:0] result;

   //recall module
   lab4task1_ASM dut (.clk, .in, .start, .shr, .zero, .result);

   //clock setup
   parameter clk_pd = 100;
   initial begin
      clk <= 0;
      forever #(clk_pd /2) clk <= ~clk;
   end //of clock setup

   //tests instance where the input has four 1's
   //and checks thoroughly through its contents
   initial begin
      in <= 8'b00011110; start <= 0; shr <= 0; @(posedge clk);
      in <= 8'b00011110; start <= 1; shr <= 1; @(posedge clk);
      in <= 8'b00011110; start <= 1; shr <= 1; @(posedge clk);
      in <= 8'b00011110; start <= 1; shr <= 1; @(posedge clk);
      in <= 8'b00011110; start <= 1; shr <= 1; @(posedge clk);
      in <= 8'b00011110; start <= 1; shr <= 1; @(posedge clk);
      in <= 8'b00011110; start <= 1; shr <= 1; @(posedge clk);
      in <= 8'b00011110; start <= 1; shr <= 1; @(posedge clk);
      in <= 8'b00011110; start <= 1; shr <= 1; @(posedge clk);
      in <= 8'b00011110; start <= 1; shr <= 0; @(posedge clk);
```

```
74          $stop;
75        end //test
76    endmodule //testbench*/
```